WO0014924

Publication Title:

ELLIPTIC CURVE CRYPTOSYSTEMS FOR LOW MEMORY DEVICES

Abstract:

Abstract of WO0014924

Each participant in a cryptographic system selects its own elliptic curve and verifies that the elliptic curve is sufficiently secure. A participant is represented by a handheld low memory device such as a smart card. A central facility is not required for key creation. The determination of whether an elliptic curve is sufficiently secure is made by counting the number of points on the curve and ensuring that this number is divisible by a prime number of at least a predetermined length. Data supplied from the esp@cenet database - Worldwide

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 9/30, G06F 7/72

A1

(11) International Publication Number: WO 00/14924

(43) International Publication Date: 16 March 2000 (16.03.00)

(21) International Application Number: PCT/US99/20411

(22) International Filing Date: 7 September 1999 (07.09.99)

(30) Priority Data:
60/099,424 8 September 1998 (08.09.98) US

(71) Applicant: CITIBANK, N.A. [US/US]; 399 Park Avenue, New York, NY 10043 (US).

(72) Inventor: CSIRIK, Janos, A.; Apt. 1, 2201 Rose Street, Berkeley, CA 94709-1430 (US).

(74) Agents: CALVARUSO, Joseph, A. et al.; Morgan & Finnegan L.L.P., 345 Park Avenue, New York, NY 10154 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: ELLIPTIC CURVE CRYPTOSYSTEMS FOR LOW MEMORY DEVICES

(57) Abstract

Each participant in a cryptographic system selects its own elliptic curve and verifies that the elliptic curve is sufficiently secure. A participant is represented by a handheld low memory device such as a smart card. A central facility is not required for key creation. The determination of whether an elliptic curve is sufficiently secure is made by counting the number of points on the curve and ensuring that this number is divisible by a prime number of at least a predetermined length.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

ELLIPTIC CURVE CRYPTOSYSTEMS FOR LOW MEMORY DEVICES

BACKGROUND OF THE INVENTION

The present invention relates to cryptosystems, and, more particularly, is directed to cryptosystems wherein a handheld device for each user of the cryptosystem selects its own elliptic curve, rather than using an elliptic curve predetermined for all users of the cryptosystem.

In a conventional elliptic curve cryptosystem, as shown in Fig. 1, a central facility selects a finite field, an elliptic curve, a generator of an appropriate subgroup of the group of points of the elliptic curve over the finite field, and determines the order of that generator. The central facility distributes these data among the participants in the cryptographic system. Each participant then selects a secret key, computes a corresponding public key, and may optionally obtain certification for its public key. The objective of the certificate is to make one party's public key available to other parties in such a way that those other parties can independently verify that the public key is valid and authentic. An advantage of the conventional system is that, while a lot of computation is required to obtain both the cardinality of the group of points of an elliptic curve over a finite field, and to find an elliptic curve for which this cardinality satisfies the security requirements, this computation need not be performed by participants - - which would be very burdensome - - as the computation is performed once by the central facility.

Conventional elliptic curve cryptosystems are used in the same applications as other public key cryptosystems, such as authentication, certification, encryption/decryption, signature generation and verification.

As shown in Fig. 2, to use the conventional elliptic curve cryptosystem, two parties wishing to communicate exchange their cryptographic data, and then proceed with their communication, such as a signature scheme or a data encryption/decryption scheme.

A serious problem with the above-described conventional elliptic curve cryptosystem is that all participants are vulnerable to an attack on the centrally selected elliptic curve and finite field. That is, the system is vulnerable to a concentrated attack on the Discrete Logarithm problem in the group defined by the centrally selected elliptic curve and finite field. Thus, there is a need to reduce the vulnerability to attack of elliptic curve cryptosystems, in particular, cryptosystems having the cryptographic functionality implemented in a small, inexpensive, low power device such as a so-called "smart card".

SUMMARY OF THE INVENTION

In accordance with an aspect of the invention, a method of selecting an elliptic curve for a cryptosystem is provided. A prime number p defining a field F_p is selected. A set of candidate elliptic curves E_i over the field F_p is selected. Then a set of modular polynomials Ψ_ℓ modulo p for a list of candidate auxiliary primes ℓ is found by a calculation in characteristic p using a stored polynomial P_ℓ. The roots modulo p of the modular polynomials Ψ_ℓ are found. Kernel polynomials h(X) based on the roots of the modular polynomials Ψ_ℓ are generated. An eigenvalue e for one of the kernel polynomials h(X) is found. A value t based on the eigenvalue e and the prime number p is obtained. The number of points of one of the candidate elliptic curves E_i over F_p is compared with the value t to make a determination whether the candidate elliptic curve is sufficiently secure. When the determination is that the

candidate elliptic curve is sufficiently secure, the candidate elliptic curve is selected for the cryptosystem.

The step of finding the set of modular polynomials Ψ_ℓ is performed by without table look-up of the modular polynomialsΨ_ℓ.

When the determination is that the candidate elliptic curve is insufficiently secure, the step of obtaining the nubmer of points is repeated for another of the candidate elliptic curves E_i.

The prime number p has about 200 bits, and the number of points of the selected elliptic curve is a product of a second prime number and a cofactor, the cofactor having up to 5 bits.

In accordance with another aspect of the invention, a method of encrypting a message M is provided, wherein an elliptic curve E is selected according to the method described above, and then the following are selected: a point P of prime order q on the selected elliptic curve E over the field of F_p, a secret positive integer m and a random positive integer k, $m < q$, $k<q$. The points $k \otimes P$ and $k \otimes (m \otimes P) = (x, y)$ on the curve E are obtained, and the point $(k \otimes P, (x * M) \bmod p)$ is obtained as the encrypted message.

In accordance with yet another aspect of the invention, a method of obtaining a digital signature for a message M is provided, wherein an elliptic curve E is selected according to the method described above, and then the following are selected: a point P of prime order q on the selected elliptic curve E over the field of F_p, a secret positive integer m and a random positive integer k, $m < q$, $k<q$. A cryptographically secure hash value d between 1 and $q - 1$ of the message M is obtained, and $k \otimes P = (x, y)$ is calculated. The pair $((x + d) \bmod q, (k - m (x + d))$

mod q) is obtained as the digital signature.

In accordance with a further aspect of the invention, a portable device for encoding information using an elliptic curve cryptosystem is provided, having means for selecting an elliptic curve by finding the roots of modular polynomials Ψ_ℓ modulo p for a list of candidate auxiliary primes ℓ and a prime number p by a calculation in characteristic p using a stored polynomial P_ℓ, and means for encoding the information using the selected elliptic curve.

In accordance with a still further aspect of the invention, a portable device for digitally signing information using an elliptic curve cryptosystem is provided, having means for selecting an elliptic curve by finding the roots of modular polynomials Ψ_ℓ modulo p for a list of candidate auxiliary primes ℓ and a prime number p by a calculation in characteristic p using a stored polynomial P_ℓ, and means for digitally signing the information using the selected elliptic curve.

It is not intended that the invention be summarized here in its entirety. Rather, further features, aspects and advantages of the invention are set forth in or are apparent from the following description and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a flowchart showing a set-up phase of a common curve elliptic curve cryptosystem;

Fig. 2 is a flowchart showing operation of a common curve elliptic curve cryptosystem;

Figs. 3A and 3B are flowcharts showing set-up and operation of a proposed user-selected curve elliptic curve cryptosystem;

Figs. 4A and 4B are flowcharts showing set-up and operation of a user-selected curve elliptic curve cryptosystem according to the present invention;

Figs. 5A-5C comprise a flowchart showing, in detail, the flowchart of Fig. 4B;

Fig. 6 is a flowchart showing selection of a suitable elliptic curve, as required in step 130 of Fig. 5A;

Fig. 7 is a flowchart showing calculation of a modular polynomial Ψ_ℓ, as required in step 220 of Fig. 5A;

Fig. 8 is a flowchart showing generation of a polynomial G_k, as required in step 780 of Fig. 7;

Fig. 9 is a flowchart showing how to obtain an eigenvalue e, as required in step 370 of Fig. 5B;

Fig. 10 is a flowchart showing how to obtain polynomials $a_s(X)$, $b_s(X)$, $c_s(X)$ and $d_s(X)$;

Fig. 11 is a flowchart showing how to obtain coefficients a_k; and

Fig. 12 is a flowchart showing how to obtain the coefficients $(-1)^i s_i$.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the present invention, each user, typically represented by a respective handheld low memory device such as a smart card, selects its own elliptic curve and verifies that the elliptic curve is sufficiently secure. It is an important aspect of the present invention that each user's device is able to independently verify the sufficiency of security of its selected elliptic curve.

It is an important aspect of the present invention that a central facility is not required during key creation but may be used during key certification. Users

wishing to communicate exchange cryptographic data, and then encrypt and decrypt as desired. Advantageously, cryptosystems according to the present invention are not vulnerable to an attack on a centrally selected elliptic curve and finite field, since such targets do not exist. Another advantage of cryptosystems according to the present invention is that a central facility cannot influence selection of cryptographic parameters, and therefore cannot disadvantage users, such as by selecting parameters with a "trapdoor" facilitating unauthorized retrieval of a user's secret key.

Practically, an elliptic curve for an elliptic curve cryptosystem is sufficiently secure when the number of points in the group of the elliptic curve, also referred to as the "order" of the elliptic curve, is divisible by a prime number of at least a predetermined length. After counting the number of points in the group of the elliptic curve, it is straightforward to assess the security of the elliptic curve. When the order is divisible by a sufficiently large prime number, then the discrete logarithm (DL) problem faced by an unauthorized user of the cryptosystem presents sufficient computational difficulty that the security of the cryptosystem is adequate.

An overview of polynomial time algorithms for determining the number of points on an elliptic curve is presented in Schoof, "Counting points on elliptic curves over finite fields", *J. de Theorie de Nombres de Bordeaux*, vol. 7, 219-254 (1995). The instant technique for finding an appropriate elliptic curve is based on the Schoof-Elkies-Atkin algorithm. Examples of algorithms are provided in Elkies, "Elliptic and modular curves over finite fields and related computational issues", in Buell et al. (ed.) *Computational Perspectives in Number Theory*, AMS, 21-76 (1998). A practical implementation of the Schoof-Elkies-Atkin algorithm is

described in Morain, "Calcul du nombre de points sur une courbe elliptique dans un corps fini: aspects algorithmiques", *J. de Theorie de Nombres de Bordeaux*, vol. 7, 255-282 (1995). Another implementation involving a match and sort method and isogeny cycles is described in Izu et al., "Efficient Implementation of Schoof's Algorithm" in *Lecture Notes on Computer Science: ASIACRYPT 98 Conference*, Beijing, Springer, 66-79 (1998).

The instant technique for determining the number of points on an elliptic curve is similar to that described in Morain's 1995 paper. As discussed further below, a modular polynomial Ψ_ℓ must be generated for each candidate auxiliary prime number ℓ.

Fig. 3A shows that, for Morain's technique, in a set-up procedure performed ahead of actual operation, the modular polynomials Ψ_ℓ for characteristic 0 are generated and stored in a TABLE. Fig. 3B shows that, for Morain's technique, during usage, the modular polynomials Ψ_ℓ are obtained via TABLE look-up, and then an appropriate elliptic curve is found.

Fig. 4A shows that, for the instant technique, in a set-up procedure, the set of modular polynomials Ψ_ℓ for ℓ belonging to a set of small primes A_s (discussed in detail below) is hard-coded in software, such as by placing the polynomials in a table. Fig. 4B shows that, for the instant technique, during usage, the modular polynomials $\Psi_\ell \bmod p$ for the ℓ in A_s are obtained by retrieving the modular polynomials Ψ_ℓ from the table and by reducing the retrieved polynomials modulo p, whereas the $\Psi_\ell \bmod p$ for ℓ not in A_s are obtained dynamically, where p is a large prime number, after which an appropriate curve is found.

The performance of Morain's technique during usage will now be compared with the performance of the instant technique during usage.

Using Morain's technique, even when a device is not performing cryptographic computing, it must keep the TABLE in memory, which consumes about 300 KB (kilobytes), for a particular security level. For the same security level, using the instant technique, when a device is not performing cryptographic computing, only executable software, including the modular polynomials Ψ_ℓ corresponding to the small primes ℓ, is kept in memory and consumes about 40 KB.

Using Morain's technique, when a device is performing cryptographic calculations, it requires about 300 KB for the TABLE and 40 KB for the executable cryptographic code, for a total requirement of 340 KB. Using the instant technique, when a device is performing cryptographic calculations, it requires about 100 KB for the dynamically calculated Ψ_ℓ and 40 KB for the executable cryptographic code, for a total requirement of about 140 KB. It is observed that since the Ψ_ℓ are not calculated in characteristic 0 during the dynamic calculation of the instant technique, only the $\Psi_\ell \bmod p$ are calculated, less memory is required than for Morain's technique, which calculates the Ψ_ℓ in characteristic 0.

Thus, it can be seen that the present technique requires dramatically less memory in a device than Morain's technique. Reduced memory requirements make it practical to use a cheaper device, which in turn makes cryptographic protection according to the present technique available to a wider range of applications.

Referring to Figs. 5A-5C, the instant technique for obtaining a suitable elliptic curve E will now be described. The steps depicted in Figs. 5A-5C are

assumed to be performed by a general purpose computer programmed in accordance with the instant technique, but may alternatively be performed by a specially designed circuit.

Let E be an elliptic curve defined using predetermined integers a_4, a_6 as follows:

$$E: \quad y^2 = x^3 + a_4 x + a_6$$

When a large odd prime p does not divide $(4a_4^3 + 27\, a_6^2)$, the elliptic curve E can be reduced to an elliptic curve over the field F_p.

Let $\#E(F_p)$ be the number of points of E over F_p, given as

$$\#E(F_p) = p + 1 - t$$

where t is an integer which satisfies

$$-2\, p^{0\,5} \leq t \leq 2\, p^{0\,5}$$

The instant technique finds t modulo several small auxiliary primes. When the product of the auxiliary primes exceeds $4\, p^{0\,5}$, the Chinese Remainder Theorem is used to recover the exact value of t, and hence the exact value of $\#E(F_p)$.

At step 110 of Fig. 5A, a prime number p having about 200 bits, hence a value around 2^{200}, is chosen. At step 120, it is determined whether $p \equiv 3 \bmod 4$; if not, then the procedure returns to step 110 and selects a different prime number p.

The instant technique proceeds with a predetermined number of candidate curves, such as 70 candidates, in parallel. For a randomly chosen elliptic curve E over F_p, the probability that $\#E(F_p) = x\, r$ for a positive integer $x \leq 30$ and a prime number r is about 3%, so approximately 70 curves must be evaluated to find a curve where the group order $\#E(F_p)$ has a large prime r which, in turn, ensures that the DL problem is sufficiently difficult. Let the predetermined number of curves be i_{MAX};

in this example $i_{MAX} = 70$. At step 130, a suitable curve E_i is found for i = 1, ... , i_{MAX}, and the following quantities dependent on E_i are also found: $j(E_i)$, a_s, b_s, c_s, and d_s.

Fig. 6 is a flowchart depicting a procedure for finding a suitable candidate elliptic curve E.

At step 600, values for the coefficients a_4 and a_6 are randomly selected in F_p.

At step 610, it is checked whether the prime number p divides $(4 a_4^3 + 27 a_6^2)$. If so, then E is not an elliptic curve when reduced modulo p and the procedure returns to step 600 to select new coefficients. If not, the procedure continues to step 620.

At step 620, the j-invariant $j(E)$ is found:

$$j(E) = 6912\, a_4^3 \,/\, (4\, a_4^3 + 27\, a_6^2) \in F_p$$

At step 640, it is checked whether the j-invariant is 0 or 1728. If so, then the procedure returns to step 600 to select new coefficients. If not, the procedure continues to step 650.

At step 650, a random point Q on E is selected, and at step 660, it is checked whether $(p + 1) \otimes Q = 0$, that is, whether $(p + 1)$ annihilates the point Q. If so, then E is probably supersingular and it is best to return to step 600 and select new coefficients. If not, then E is definitely not supersingular and the procedure continues to step 670. If $(p + 1) \otimes Q = 0$, then steps 650 and 660 may be repeated for another randomly chosen point Q, to decrease the likelihood of rejecting a curve that is not supersingular.

At step 670, values are initialized for the Chinese Remainder count of the trace t. The modulus M for E with respect to known t is set to 1. The value T such

that $t \equiv T \bmod M$ is set to 0.

At step 690, expressions modulo p are found for the polynomials $a_s(X)$, $b_s(X)$, $c_s(X)$ and $d_s(X)$ for $s \leq R$ as follows, where the upper bound R=11 is large enough for the set of candidate auxilary prime numbers ℓ used here. Fig. 10 is a detailed flowchart for the processing that occurs at step 690 of Fig. 6.

At step 1010 of Fig. 10, the following terms are initialized:

$$w(X) = X^3 + a_4 X + a_6$$

$$f_1(X) = 1$$

$$f_2(X) = 2$$

$$f_3(X) = 3X^4 + 6 a_4 X^2 + 12 a_6 X - a_4^2$$

$$f_4(X) = 4X^6 + 20 a_4 X^4 + 80 a_6 X^3 - 20 a_4^2 X^2 - 16 a_4 a_6 X - 4 a_4^3 - 32 a_6^2$$

At step 1020, polynomials are determined for s = 2 as follows:

$$a_2(X) = 4 X w(X) - f_3(X)$$

$$b_2(X) = 4 w(X)$$

$$c_2(X) = f_4(X) / 4$$

$$d_2(X) = 8 w(X)^2$$

At step 1030, a counter n is set to a value of 5.

At step 1040, it is checked whether n is even.

If the result of the check at step 1040 is that n is even, then at step 1050, m is set to n/2. At step 1060, the expression f_n is set to $f_m (f_{m+2} f_{m-1}^2 - f_{m-2} f_{m+1}^2) / 2$, and processing proceeds to step 1110.

If the result of the check at step 1040 is that n is odd, then at step 1070, m is set to (n - 1)/2. At step 1080, it is checked whether m is even. If m is even, then at step 1090, f_n is set to $w^2 f_{m+2} f_m^3 - f_{m-1} f_{m+1}^3$, and processing proceeds to step 1110. If m is odd, then at step 1100, f_n is set to $f_{m+2} f_m^3 - w^2 f_{m-1} f_{m+1}^3$, and processing proceeds to step 1110.

At step 1110, the counter n is incremented. At step 1120, it is checked whether n = R + 3. If not, then processing returns to step 1040.

If the result of the check at step 1120 is positive, then at step 1130, s is set to 3.

At step 1140, it will be appreciated that s is odd and in the range $2 < s \leq R$. Polynomials are evaluated as follows:

$$a_s(X) = X f_s(X)^2 - w(X) f_{s-1}(X) f_{s+1}(X)$$

$$b_s(X) = f_s(X)^2$$

$$c_s = f_{s+2}(X) f_{s-1}(X)^2 - f_{s-2}(X) f_{s+1}(X)^2$$

$$d_s(X) = 4 f_s(X)^3$$

The polynomials $a_S(X)$, $b_S(X)$, $c_S(X)$ and $d_S(X)$ are stored, for retrieval at step 920, discussed below.

At step 1150, s is incremented by 2, that is, to be the next odd number. At step 1160, it is checked whether s > R. If so, then processing terminates. If not, then processing returns to step 1140.

Returning to Fig. 6, at step 695, the procedure is completed and a suitable *E* has been found. It will be appreciated that the procedure of Fig. 6 is repeated to obtain each of the candidate curves *E*.

Returning to Fig. 5A, at step 160, a temporary value g is initialized to "1".

At step 170, the temporary value g is used as an index into the set A of auxiliary primes {A[1], A[2], ..., A[36]}:

A = {3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 41, 47, 59, 71, 53, 61, 79, 83, 89, 101, 73, 131, 103, 107, 109, 97, 113, 151, 167, 127, 179, 139, 191, 149, 137, 173}

At this point, g = 1, so the first of the auxiliary primes is obtained and used as the value for a candidate auxiliary prime ℓ. After the first execution of step 170, $\ell = 3$.

At step 200, the modular polynomial Ψ_ℓ for the auxiliary prime ℓ currently being evaluated is obtained. If ℓ is one of the first eight of the auxiliary primes, then Ψ_ℓ is obtained by look up in Table 1.

TABLE 1

auxiliary prime ℓ	**modular polynomial Ψ_ℓ (F, J)**
3	$F^4 + (-J + 792) F^3 + (-36 J + 221400) F^2 + (1916 J + 24690528) F + (J^2 + 50976 J + 803894544)$
5	$F^6 + (-J + 780) F^5 + (-30 J + 218940) F^4 + (310 J + 25968800) F^3 + (13700 J + 1177897200) F^2 + (38424 J + 22576632000) F + (J^2 - 614000 J + 155720872000)$
7	$F^8 + (- J + 776) F^7 + (- 28J + 217756) F^6 + (21J + 26195512) F^5 + (6328J + 1276406726) F^4 + (39361J + 31050881848) F^3 + (- 240492J + 404938789276) F^2 + (- 2176581J + 2721214073864) F + (J^2 - 1711008J + 7427483226241)$

auxiliary prime ℓ	modular polynomial Ψ_ℓ (F, J)
11	$F^{12} + (-J + 684)F^{11} + (55J + 157410) F^{10} + (-1188J + 12515580) F^9 + (12716J + 75763215) F^8 + (-69630J + 76077144) F^7 + (177408J - 207606564) F^6 + (-133056J - 34321320) F^5 + (-132066J + 418524975) F^4 + (187407J - 477130500) F^3 + (-40095J + 270641250) F^2 + (-24300J - 82012500) F + (J^2 + 6750J + 11390625)$
13	$F^{14} + (-J + 772) F^{13} + (-26J + 216424) F^{12} + (-156J + 26333528) F^{11} + (1508J + 1359640022) F^{10} + (21658J + 39120460496) F^9 + (39624J + 716780223796) F^8 + (-612742J + 8956723925032) F^7 + (-3355976J + 79070093432161) F^6 + (454779J + 500196729175884) F^5 + (43741490J + 2260671730897788) F^4 + (95939974J + 7142292018579744) F^3 + (-41335164J + 15009662255513328) F^2 + (-291162600J + 18874201488396480) F + (J^2 - 174668400J + 10755802087387200)$
17	$F^{18} + (-J + 690) F^{17} + (51J + 160191) F^{16} + (-1105J + 12849212) F^{15} + (13243J + 77940903) F^{14} + (-95659J - 24306702) F^{13} + (424065J + 489756655) F^{12} + (-1110355J + 856070496) F^{11} + (1454945J + 247945272) F^{10} + (-73746J - 4127455840) F^9 + (-2450210J + 10326614640) F^8 + (3131026J - 15993234432) F^7 + (-1104830J + 18158824448) F^6 + (-1073992J - 15889021440) F^5 + (1392232J + 10788499200) F^4 + (-557600J - 5622784000) F^3 + (-2720J + 2154240000) F^2 + (67200J - 537600000) F + (J^2 - 16000J + 64000000)$

auxiliary prime ℓ	**modular polynomial Ψ_ℓ (F, J)**
19	$F^{20} + (-J + 664) F^{19} + (76J + 143260) F^{18}$ $+ (-2622J + 9204360) F^{17} + (54454J - 176115066) F^{16}$ $+ (-761425J + 1108178952) F^{15} + (7598556J - 1742337316) F^{14}$ $+ (-55989713J - 13420942600) F^{13}$ $+ (310967414J + 7967345585) F^{12}$ $+ (-1317638334J - 133492721376) F^{11}$ $+ (4284347658J - 271425795648) F^{10}$ $+ (-10696404825J + 1738318231104) F^{9}$ $+ (20413753140J - 3257912161280) F^{8}$ $+ (-29485216120J + 528231178240) F^{7}$ $+ (31694225470J + 10718241992704) F^{6}$ $+ (-24698209440J - 26958821326848) F^{5}$ $+ (13397395220J + 36334713176064) F^{4}$ $+ (-4738229120J - 31060143636480) F^{3}$ $+ (973578240J + 16944463872000) F^{2}$ $+ (-91238400J - 5430382166016) F$ $+ (J^{2} + 1769472J + 782757789696)$
23	$F^{24} + (-J + 720) F^{23} + (23J + 179952) F^{22}$ $+ (-161J + 17282016) F^{21} + 441081120F^{20}$ $+ (3864J + 5678198784) F^{19} + (-5681J + 45492865088) F^{18}$ $+ (-46644J + 252605710080) F^{17}$ $+ (53084J + 1038071734272) F^{16}$ $+ (393024J + 3294356631552) F^{15}$ $+ (19136J + 8309302456320) F^{14}$ $+ (-1978368J + 16991995871232) F^{13}$ $+ (-2689666J + 28563290271744) F^{12}$ $+ (2882544J + 39839110889472) F^{11}$ $+ (11625488J + 46370418130944) F^{10}$ $+ (11002464J + 45154515419136) F^{9}$ $+ (-3833824J + 36762400456704) F^{8}$ $+ (-19783680J + 24919460020224) F^{7}$ $+ (-21906304J + 13946021740544) F^{6}$ $+ (-11787776J + 6353857806336) F^{5}$ $+ (-1554432J - 2304837156864) F^{4}$ $+ (2213888J + 642483486720) F^{3}$ $+ (1648640J + 129654325248) F^{2}$ $+ (516096J + 16911433728) F$ $+ (J^{2} + 65536J + 1073741824)$

If ℓ is one of the remaining auxiliary primes, i.e., not one of the first eight auxiliary

primes, then $\Psi_\ell \bmod p$ is obtained by computation, as described in Fig. 7.

At step 710, the value of a polynomial P_ℓ is obtained by look-up from Table 2. Let ν be the degree of P_ℓ, that is, -1 times the smallest exponent occurring in J. The first column of Table 2 indicates the particular prime number ℓ under consideration. The second column of Table 2 indicates the number of coefficients in F_p which must be stored in connection with the polynomial P_ℓ (J), which is given in the third column of Table 2.

TABLE 2

ℓ	$8\ell\nu + \ell + 3$	P_ℓ (J)
29	264	$J + 11$
31	282	$J + 1$
41	372	$J - 5$
47	426	$J + 9$
59	534	$J + 24$
71	642	$J - 33$
53	904	$J^2 - 3J + 26$
61	1040	$J^2 - 23J - 1$
79	1346	$J^2 + 14J - 1$
83	1414	$J^2 + 7J - 2$
89	1516	$J^2 + 26J - 17$
101	1720	$J^2 + 27J - 13$
73	1828	$J^3 + 32J^2 - 30J + 1$
131	2230	$J^2 - 47J - 51$
103	2578	$J^3 + 34J^2 - 7J - 2$
107	2678	$J^3 + 16J^2 - 32J + 11$
109	2728	$J^3 - 51J^2 + 52J$
97	3204	$J^4 + 32J^3 + 42J^2 - 24J - 2$
113	3732	$J^4 - 37J^3 + 24J^2 - 3J - 36$

ℓ	$8\ell\nu+\ell+3$	P_ℓ (J)
151	3778	$J^3 + 34\,J^2 - 7\,J - 1$
167	4178	$J^3 - 60\,J^2 + 3\,J - 14$
127	4194	$J^4 - 54\,J^3 - 41\,J^2 - 32\,J - 2$
179	4478	$J^3 - 83\,J^2 + 18\,J - 62$
139	4590	$J^4 - 56\,J^3 - 18\,J^2 + 40\,J + 1$
191	4778	$J^3 + 60\,J^2 - 25\,J + 56$
149	4920	$J^4 + 5\,J^3 - 61\,J^2 + 48\,J - 57$
137	5620	$J^5 - 20\,J^4 - 23\,J^3 + 53\,J^2 + 65\,J + 52$
173	5712	$J^4 - 34\,J^3 - 60\,J^2 - 74\,J - 22$

At step 720, the coefficients $a_k \in F_\ell$ are obtained. Fig. 11 is a flowchart for obtaining the coefficients a_k.

At step 1210 of Fig. 11, the truncated power series X is obtained by considering modulo ℓ the power series

$$P_\ell\,(j(q))\;\overline{\eta}\,(q)\;\overline{\eta}\,(q^\ell) \equiv \sum_{k=-\nu}^{2\ell\nu-\nu} a_k q^k \;+ O(q^{(2\nu+1)\ell+1}) \qquad (\mathrm{mod}\ \ell)$$

and dropping all powers of q with an exponent of at least $2\ell\nu - \nu + 1$.

At step 1220, k is set to $-\nu$.

At step 1230, the coefficients a_k (which are not to be confused with the polynomials a_s)

are obtained by multiplying the terms on the left hand side modulo ℓ and reading off the resulting coefficients. The polynomial P_ℓ was obtained in step 710. The term $j(q)$ is obtained from:

$$j(q) \;=\; 1728E_4(q)^3/(E_4(q)^3 - E_6(q)^2)$$

$$= q^{-1} + 744 + 196884q + 21493760q^2 + \ldots$$

For any integer n, let the function $\sigma_k(n)$ denote the sum of the kth powers of the positive divisors of n. The q-series used in the above equation are given as:

$$E_4(q) = 1 + 240 \sum_{\pi=1}^{\infty} \sigma_3(n) q^n$$

$$= 1 + 240\, q + 2160\, q^2 + 6720\, q^3 + \ldots$$

$$E_6(q) = 1 - 504 \sum_{\pi=1}^{\infty} \sigma_5(n) q^n$$

$$= 1 - 504\, q - 166532\, q^2 - 122976\, q^3 - \ldots$$

The term $\overline{\eta}\ (q)$ is obtained from

$$\overline{\eta}\ (q) = \prod_{n=1}^{\infty} (1 - q^n)$$

$$= \sum_{k=-\infty}^{\infty} (-1)^k q^{(3k^2 + k)/2}$$

$$= 1 - q - q^2 + q^5 + \ldots$$

Although the q-series for $E_4(q\ )$, $E_6(q)$, $j(q)$ and $\overline{\eta}\ (q)$ do not depend on ℓ, their coefficients increase quickly and are only needed modulo ℓ or modulo p. Therefore, it is advantageous to compute them each time they are needed, rather than storing them. In a variation, only $1/\overline{\eta}\ (q)$ modulo p is computed and stored, since it is used for each auxiliary prime ℓ.

At step 1240, it is checked whether $k = 2\ell\ \nu - \nu$. If yes, then processing in Fig. 11 terminates. If not, then at step 1250, k is incremented and processing returns to step 1230.

Returning to Fig. 7, at step 730, the coefficients b_k (which are not to be

confused with the polynomials b_s) are obtained. For each k between $-v$ and $2\ell v$-v, the coefficient b_k is the least absolute remainder of a_k modulo ℓ, that is, the integer with the smallest possible absolute value that reduces to a_k modulo ℓ.

At step 740, the q-series for f is obtained:

$$f(q) = \left(\sum_{k=-v}^{(2\ell v - v)} b_k q^k + O(q^{(2v+1)\ell+1}) \right) / (\bar{\eta}(q)\bar{\eta}(q^{\ell})) \qquad \text{modulo } p$$

At step 750, the q-expansions of $f, f^2, \ldots, f^{\ell}$ are obtained and used to define $a(n,k)$:

$$(f(q))^k = \sum_n a(n,k)\, q^n$$

At step 760, the terms $s_k(q)$, for $1 \le k \le l$ are obtained. For each $1 \le k \le l$, let

$$s_k(q) = \sum_n \ell a(\ell n, k) q^n$$

At step 765, the terms $c_k(q)$, for $1 \le k \le l$ are obtained. For each $1 \le k \le l$, let

$$c_k(q) = -\left(s_k(q) + \sum_{r=1}^{k-1} c_{k-r}(q) s_r(q) \right) / k$$

At step 770, the initial and final terms of $C(q)$ are set:

$$C_1(q) = -f + c_1(q)$$

$$C_{\ell+1}(q) = -f\, c_{\ell}(q)$$

At step 775, the terms $C_k(q)$ for each $2 \le k \le \ell$ are obtained:

$$C_k(q) = -f\, c_{k-1}(q) + c_k(q).$$

At step 780, the polynomials G_k for $1 \le k \le \ell + 1$ are obtained. For each $1 \le k \le \ell + 1$, there is a polynomial G_k such that $G_k(j(q)) \equiv C_k(q) \bmod p$. Fig. 8 is a flowchart of a procedure for determining G_k.

At step 810 of Fig. 8, set $z = c_k(q)$. At step 820, set t = order (z), that is,

$$t = -\min\{n: \text{coeff}(q^n \text{ in } z) \neq 0\}$$

At step 830, set R = 0 and b = t. The value R is used to accumulate G_k. The value b is decremented so as to accumulate G_k terms for each power of z.

At step 840, set $R = R + J^b \text{coeff}(q^{-b} \text{ in } z)$. At step 850, set $z = z - \text{coeff}(q^{-b} \text{ in } z)\,(j(q))^b$.

At step 860, determine whether b = 0. If not, then there are additional powers of z to be evaluated, so at step 870, b is decremented and the procedure returns to step 840.

If b = 0 then all powers of z have been evaluated, and the procedure returns with $G_k = R$.

Returning to Fig. 7, at step 790, the modular polynomial $\Psi_\ell \bmod p$ is generated based on the polynomials G_k.

$$\Psi_\ell(F, J) = F^{\ell+1} + \sum_{i=1}^{\ell} G_i(J)\, F^{\ell+1-i}$$

Returning to Fig. 5A, at step 210, a counter i is set to 1. The counter i is used to index the candidate elliptic curves under evaluation. Of course, other numbers of elliptic curves could be evaluated in parallel, or the elliptic curves could be evaluated serially, corresponding to $i_{MAX} = 1$.

At step 220, the roots f in the field F_p of the expression $\Psi_\ell(j(E_i), f) = 0$ are obtained. These roots may be obtained using Berlekamp's second algorithm, as described at H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, 1993, pages 123-132. Let the set of roots be $\{f_1, \ldots, f_{d_{max}}\}$ where d_{max} is the number of distinct roots f.

At step 240, for each of the roots f_d, $d = 1$ to d_{max} (where d_{max} is from step 220), find all roots $\tilde{j} \in F_p$ of $\Psi_\ell(\tilde{j}, f_d) = 0$. These roots may also be obtained using Berlekamp's second algorithm, as discussed above.

At step 270, any entries equal to 0 or 1728 in the lists of roots $\tilde{j}$ are deleted.

Turning to Fig. 5B, at step 300, for the first of the pairs of roots $(f, \tilde{j})$, values are obtained for the variables $\tilde{a}_4$, $\tilde{a}_6$ and p_1 via the following intermediate calculations:

$$E_4 = -48a_4$$

$$E_6 = 864a_6$$

$$f = \frac{E_6}{E_4} j \frac{\Psi_2(f,j)}{\Psi_1(f,j)}$$

$$Q = \frac{f'}{\tilde{j}} \frac{1}{\ell} \frac{\Psi_1(f,\tilde{j})}{\Psi_2(f,\tilde{j})}$$

$$\tilde{E}_4 = \frac{\tilde{j}}{\tilde{j}-1728} Q^2$$

$$\tilde{E}_6 = \tilde{E}_4 Q$$

$$t_1 = \frac{1}{\Psi_1(f,j)} \left(-f' \quad \Psi_{11}(f,j) + 2j\Psi_{12}(f,j)\frac{E_6}{E_4} - \frac{E_6^2}{fE_4^2}(j\Psi_2(f,j) + j^2\Psi_{22}(f,j)) \right)$$

$t_2 =$

$$\frac{1}{\Psi_1}(f,\tilde{j}) \left(-f'\Psi_{11}(f,\tilde{j}) + 2\ell_{\tilde{j}}\Psi_{12}(f,\tilde{j})\frac{\tilde{E}_6}{\tilde{E}_4} - \ell^2 \frac{\tilde{E}_6^2}{f'\tilde{E}_4^2}(\tilde{j}\Psi_2(f,\tilde{j}) + \tilde{j}^2\Psi_{22}(f,\tilde{j})) \right)$$

$$t_3 = \frac{E_6}{3E_4} - \frac{E_4^2}{2E_6}$$

$$t_4 = \ell\left(\frac{\widetilde{E}_6}{3\widetilde{E}_4} - \frac{\widetilde{E}_4^2}{2\widetilde{E}_6}\right)$$

$$p_1 = \ell\frac{t_2 + t_4 - t_1 - t_3}{4}$$

$$\widetilde{a}_4 = -\ell^4\widetilde{E}_4 / 48$$

$$\widetilde{a}_6 = \ell^6\widetilde{E}_6 / 864$$

The values for all intermediate values may be discarded, that is, only the values for $\widetilde{a}_4$, $\widetilde{a}_6$ and p_1 are retained.

At step 310, the kernel polynomial h(X) of degree $d = (\ell - 1)/2$ is determined based on the values for $\widetilde{a}_4$, $\widetilde{a}_6$ and p_1 obtained at step 300. Figure 12 is a flowchart for the processing that occurs at step 310 of Fig. 5B.

At step 1310 of Fig. 12, the following values are set:

$$p_0 = d$$

$$p_2 = ((1 - 10\,d)\,a_4 - \widetilde{a}_4)/\,30$$

$$p_3 = ((1 - 28\,d)\,a_6 - 42\,p_1\,a_4 - \widetilde{a}_6)\,/\,70$$

$$c_1 = 6\,p_2 + 2\,a_4\,d$$

$$c_2 = 10\,p_3 + 6\,a_4\,p_1 + 4\,a_6\,d$$

At step 1320, a small positive integer S is selected that determines the number of extra terms which will be carried, such as S = 3.

At step 1330, for each $2 \le r \le d - 1 + S$, the term c_{r+1} is obtained as follows:

$$c_{r+1} = \frac{3\sum_{n=1}^{r-1} c_n c_{r-n} - (2r-1)(r-1)a_4 c_{r-1} - (2r-2)(r-2)a_6 c_{r-2}}{(r-1)\ (2r+5)}$$

At step 1340, for each $3 \leq n \leq d - 1 + S$, the term p_{n+1} is obtained as follows:

$$p_{n+1} = \frac{1}{4n+2}\left(c_n - (4n-2)\, a_4\, p_{n-1} - (4n-4)\, a_6 p_{n-2}\right)$$

These p_{n+1} terms are power sums of the roots of $h(X)$.

At step 1350, s_0 is set to be 1.

At step 1360, for $1 \leq i \leq d + S$, the term s_i is obtained as follows:

$$s_i = \frac{-1}{i}\sum_{k=1}^{i}(-1)^k p_k s_{i-k}$$

Returning to Fig. 5B, at step 330, the procedure checks whether the result obtained at step 310 is valid. Specifically, a check is made as to whether $s_{d+1} = s_{d+2} = \ldots = s_{d+S} = 0$ for the terms obtained at step 1360 of Fig. 12.

If the result of the check at step 330 of Fig. 5B fails, that is, it is not the case that $s_{d+1} = s_{d+2} = \ldots = s_{d+S} = 0$, then, at step 340, the procedure determines whether there are any untried root pairs (f, $\tilde{j}$). If so, then at step 350, the next of the pairs (f, $\tilde{j}$) is selected, and the procedure returns to step 300. If all root pairs (f, $\tilde{j}$) have been tried, then the elliptic curve E_i being evaluated is not acceptable, and the procedure moves to step 400.

If the result of the check at step 330 is successful, that is, it is the case that $s_{d+1} = s_{d+2} = \ldots = s_{d+S} = 0$, then the procedure moves to step 360, and obtains the kernel polynomial $h(X)$ as follows:

$$h(X) = \sum_{i=0}^{d}(-1)^i s_i X^{d-i}$$

At step 370, the eigenvalue e based on the kernel polynomial $h(X)$ is

obtained. Fig. 9 is a flowchart illustrating a procedure for finding the eigenvalue e.

At step 905, $h(X)$ is factored modulo ℓ using Berlekamp's algorithm.

At step 910, one of the factors of $h(X)$ is henceforth used instead of $h(X)$. In one embodiment, a factor of smallest degree is selected. In other embodiments, any factor of suitably small degree is selected.

At step 915, the value of ℓ is used to obtain a value for s, by lookup in Table 3.

TABLE 3

ℓ	s
3, 5, 7, 11, 13, 19, 23, 29, 47, 59, 71, 53, 61, 79, 83, 101, 131	2
103, 107, 167, 179, 139, 191, 149, 173	2
17, 31, 89, 113, 127, 137	3
73, 97, 151	5
41	7
109	11

At step 920, the polynomials $a_s(X)$, $b_s(X)$, $c_s(X)$, $d_s(X)$ corresponding to the elliptic curve under consideration, as found in step 690, are retrieved.

At step 925, the degree of $h(X)$ is obtained. If the result is even, the next step is step 930. If the result is odd, the next step is step 960.

At step 930, parameters are initialized as follows:

$$Q_1(X) = X^p \bmod h(X)$$

$$Q_2(X) = (X^3 + a_4 X + a_6)^{(p-1)/2} \bmod h(X)$$

$$P_1(X) = X \bmod h(X)$$

$$P_2(X) = 1$$

$$e = 1$$

At step 935, a check is made as to whether ($P_1(X)$, $P_2(X)$) = ($Q_1(X)$, $\pm Q_2(X)$).

If the check at step 935 is negative, then at step 940, the parameters are simultaneously updated as follows, that is, the new $P_1(X)$ and $P_2(X)$ are each based on the previous $P_1(X)$:

$$P_1(X) = \frac{a_s(P_1(X))}{b_s(P_1(X))} \bmod h(X)$$

$$P_2(X) = P_2(X) \frac{c_s P_1(X))}{d_s(P_1(X))} \bmod h(X)$$

$$e = e\, s \bmod \ell$$

Step 940 is repeated, at most (ℓ-1)/2 times, until the condition ($P_1(X)$, $P_2(X)$) = ($Q_1(X)$, $\pm Q_2(X)$) is true. When the condition is true, the desired eigenvalue e has been found.

At step 945, a check is made as to whether $P_2(X) = Q_2(X)$. If so, then at step 950, the desired eigenvalue is e. Otherwise, at step 955, the desired eigenvalue is determined as -e. The desired eigenvalue is then used at step 380 of Fig. 5B.

At step 960 of Fig. 9, parameters are initialized as follows:

$$Q_1(X) = X^p \bmod h(X)$$

$$P_1(X) = (X \bmod h(X))$$

$$e = 1$$

At step 965, a check is made as to whether $P_1(X) = Q_1(X)$.

If the check at step 965 is negative, then step 970, the parameters are updated as follows:

$$P_1(X) = \frac{a_s(P_1(X))}{b_s(P_1(X))} \bmod h(X)$$

$$e = e\ s \bmod \ell$$

Step 970 is repeated, at most (ℓ-1)/2 times, until the condition $P_1(X) = Q_1(X)$ is true. When the condition is true, the desired eigenvalue e has been found.

At step 975, the desired eigenvalue is $e^{s(e)}$ (r/ℓ) e, where r is the resultant of $h(X)$ and $w(X) = (X^3 + a_4 X + a_6)$ and s(e) is the semi-order of e modulo ℓ, that is, the smallest positive n such that $e^n \equiv \pm 1$ (modulo ℓ). A resultant is defined in Cohen, page 118, definition 3.3.2, and may be computed using Cohen, page 121, algorithm 3.3.7.

Returning to Fig. 5B, at step 380, the value $t = e + (p/e)$ modulo ℓ is obtained. An extended Euclidean algorithm procedure for finding t is given in Cohen, pages 12-19, particularly page 16, algorithm 1.3.6.

At step 390, with $x \equiv T_i \bmod M_i$ and $x \equiv t \bmod \ell$, use the Chinese Remainder Theorem to find $x \equiv F \bmod \ell M_i$. The Chinese Remainder Theorem is described in Cohen, pages 19-21.

The value F is chosen to have a minimum absolute value by subtracting ℓM_i from the least non-negative remainder modulo ℓM_i if the least non-negative remainder is larger than $\ell M_i/2$.

At step 395, values are reset as follows: T_i is set to be F, and M_i is set to be ℓM_i. This completes evaluation of the current elliptic curve E_i.

Turning to Fig. 5C, at step 400, it is checked whether there are any more elliptic curves to be evaluated. If so, then at step 410, the counter i is incremented,

thereby selecting the next elliptic curve, and the procedure returns to step 220.

If, at step 400, it is determined that there are no more elliptic curves to evaluate, then at step 420 it is checked whether there are any more candidate auxiliary primes to be evaluated. If so, then at step 430, the counter g is incremented, thereby selecting the next candidate auxiliary prime, and the procedure returns to step 170.

If, at step 420, it is determined that there are no more candidate auxiliary primes to evaluate, then at step 440, a counter i is initialized. Once again, the counter i is used to indicate which of the possible elliptic curves is being considered.

At step 450, it is checked whether $M_i > 4\ p^{0\ 5}$, that is, whether the bound for M_i has been reached. If not, then at step 460, it is checked whether $i = i_{MAX}$, that is, whether there are any more elliptic curves. If there are, then at step 470, i is incremented and the procedure returns to step 450. If not, then all candidate elliptic curves for the originally chosen prime number *p* have failed to yield an acceptable elliptic curve, so the procedure returns to step 110 to pick a new prime number *p*.

If, at step 450, it is determined that $M_i > 4\ p^{0\ 5}$, then at step 480, the value g is set to $p + 1 - T_i$, and at step 490, the largest $x \leq 32$ such that x divides g is found. This largest x is referred to as the cofactor β. The value 32 is equal to 2^5, with the value 5 being a second security parameter.

There are two main security parameters in the instant procedure. The first security parameter is embodied in step 110, and is the length in bits of the prime number *p*. The second security parameter is embodied in step 490, and is the logarithm to the base 2 of the largest small factor, rounded up to the nearest power of two, which divides g. This second security parameter is referred to as the

maximum allowable length of the cofactor β. The difference between the two security parameters, in this case, 200 - 5 = 195, is a measure of the security of an elliptic curve chosen by the instant procedure, with a larger difference value indicating higher security.

At step 500, it is determined whether g/x is prime, such as by using a probabilistic compositeness test wherein if g/x can be proved to be composite, then g/x is not prime, and if the proof of compositeness for g/x fails, then g/x is assumed to be prime. A probabilistic compositeness test is described in A.K. Lenstra and H.W. Lenstra, Jr., "Algorithms in Number Theory" in *Handbook of Theoretical Computer Science*, J. van Leeuwen ed., pages 675-677 and 706-715, Elsevier Science 1990, the disclosure of which is hereby incorporated by reference. If the quotient g/x is not prime, then the procedure moves to step 460 to check the next elliptic curve.

If the quotient g/x is prime, then the procedures moves to step 505 to check if the present elliptic curve is insecure, that is, if g/x divides p^k-1 for a positive integer k that is "too small" so that a sub-exponential attack on F_{p^k} would be faster than a square-root attack on $E(F_p)$, which corresponds to

$$\exp((1.923 + o(1))(k \log(p))^{1/3} (\log(k \log(p)))^{2/3}) < p^{1/2}$$

If it is determined at step 505 that the present elliptic curve is insecure, then the procedure moves to step 460 to check the next elliptic curve.

If the present elliptic curve is determined to be secure at step 505, then an acceptable elliptic curve E_i has been found, and the procedure is finished.

In a modification, after step 500, if the quotient g/x is prime, rather than immediately terminating at step 510, the modified procedure collects the prime

quotients for all the elliptic curves being evaluated, then chooses the curve with the largest quotient g/x, because that curve will be the most secure.

In another modification, instead of step 200 in Fig. 5A, the Ψ_l can be found by table look-up, as is done by Morain (see page 264 Remarque), with the calculations in Fig. 7 done in characteristic 0, rather than modulo p, and at step 370 as soon as $4p^{1/2}/M_i$ is sufficiently small, g may be found using a baby step-giant step approach, described in Cohen at pages 235-238, or rho-like methods, described in Cohen at pages 419-422

In another modification, the technique of calculating the modular polynomials $\Psi_\ell \bmod p$ is combined with Morain's method of the isogeny cycles to allow the calculation to be carried out using fewer auxiliary primes.

An example of practicing the present technique will now be provided.

At step 110 of Fig. 5A, a prime is selected. For this example, a very short prime number, p = 9883, is chosen. It will be understood that, in practice, a much longer (larger) prime number is required for sufficient security.

At step 120, it is determined that 9883 = (4)(2470) + 3, so that $p \equiv 3 \pmod 4$ is true.

At step 130, for this example, $i_{max} = 1$ is chosen. In practice, a larger value would be used. To find an elliptic curve E_1, at step 600 of Fig. 6, the values $a_4 = 123$ and $a_6 = 765$ are chosen. At step 610, the expression

$$\frac{4(123)^3 + 27(765)^2}{9883} = \frac{23244543}{9883}$$

is evaluated and determined to not be an integer. At step 620, j(E) is obtained:

$$j(E) = \frac{6912\ (123)^3}{4\ (123)^3 + 27\ (765)^2} = \frac{476381952}{860909}$$

At step 640, neither of the conditions are true. At step 650, a value Q = (235, 2241) is selected; this is a point on E. At step 660, the following calculation is made:

$$(9883 + 1) \otimes (235, 2241) = (1057, 6231) \neq 0$$

At step 670, M=1, T=0 and t=0 mod 1. To perform step 690 of Fig. 6, processing moves to step 1010 of Fig. 10.

At step 1010 of Fig. 10, the following terms are set:

$$w(X) = X^3 + 123X + 765$$

$$f_1(X) = 1$$

$$f_2(X) = 2$$

$$f_3(X) = 3X^4 + 738X^2 + 9180a_6X + 4637$$

$$f_4(X) = 4X^6 + 2460X^4 + 1902X^3 + 3793X^2 + 6579X + 9399$$

At step 1020, the following expressions are obtained:

$$a_2(X) = X^4 + 9637X^2 + 3763X + 5246$$

$$b_2(X) = 4X^3 + 492X + 3060$$

$$c_2(X) = X^6 + 615X^4 + 5417X^3 + 3419X^2 + 9057X + 9762$$

$$d_2(X) = 8X^6 + 1968X^4 + 2357X^3 + 2436X^2 + 3304X + 7141$$

Processing proceeds through steps 1030 and 1040. At step 1070, m = (5-1)/2 = 2 is obtained. Via step 1080, processing goes to step 1090 and generates the following expression:

$$f_5(X) = 5X^{12} + 7626X^{10} + 4093X^9 + 2618X^8 + 145X^7 + 4117X^6 + 2635X^5 + 2327X^4 + 2640X^3 + 9386X^2 + 3207X + 6568$$

At step 1110, n is incremented to n = 6. At step 1120, it is checked whether

6 = 10 + 3; since it is not, processing returns to step 1040, thence to step 1050 to set m = 6/2 = 3, and then to step 1060 to obtain:

$$f_6(X) = 6X^{16} + 7829X^{14} + 328X^{13} + 5633X^{12} + 2016X^{10} + 1819X^9 + 391X^8 + 8771X^7 + 1126X^6 + 7115X^5 + 5246X^4 + 4414X^3 + 8147X^2 + 7098X + 432$$

At step 1110, n is incremented to n = 7. Details of iterations until n is incremented to n = 13 are omitted for brevity. At step 1130, s is set to s = 3. At step 1140, the following expressions are obtained:

$$a_3(X) = X^9 + 8407X^7 + 5624X^6 + 9135X^5 + 4927X^4 + 7552X^3 + 3567X^2 + 1736X + 9178$$

$$b_3(X) = 9X^8 + 4428X^6 + 5665X^5 + 9135X^4 + 87X^3 + 5235X^2 + 3158X + 6244$$

$$c_3(X) = 4X^{12} + 941X^{10} + 1156X^9 + 6573X^8 + 8607X^7 + 7575X^6 + 9293X^5 + 8824X^4 + 4431X^3 + 7342X^2 + 6765X + 9442$$

$$d_3(X) = 108X^{12} + 640X^{10} + 3140X^9 + 5958X^8 + 3132X^7 + 3565X^6 + 4774X^5 + 6714X^4 + 461X^3 + 3319X^2 + 2006X + 4718.$$

At step 1150, s is incremented by 2 to s = 5. Details of iterations until s is incremented to s = 11 are omitted for brevity. At step 1170, processing returns to step 695 of Fig. 6.

At step 695 of Fig. 6, processing returns to step 160 of Fig. 5A.

At step 160 of Fig. 5A, g is set to g = 1. At step 170, ℓ is set to $\ell = 3$. At

step 200, the modular polynomial Ψ_3 is obtained from Table 1. At step 210, i is set to i = 1. At step 220, the roots of the following expression are found:

$$0 = F^4 + 9420F^3 + 8209F^2 + 5805F + 7290.$$

Specifically, there is only one root in $F_{9883} = F_p$, f = 370. At step 240, the roots of the following expresion are found:

$$0 = \tilde{J}^2 + 9380\tilde{J} + 5008.$$

Specifically, the roots of $\tilde{j} \in F_{9883}$ are 1255 and 9131. At step 270, neither of the roots of $\tilde{j}$ are deleted. At step 300 of Fig. 5B, the pair (f, $\tilde{j}$) = (370, 9131) is selected. To calculate $\tilde{a}_4, \tilde{a}_6$ and p_1, processing as described above with regard to Fig. 5B, step 300, is executed, to obtain:

$$E_4 = 3979$$

$$E_6 = 8682$$

$$f' = 446$$

$$Q = 8595$$

$$\tilde{E}_4 = 5314$$

$$\tilde{E}_6 = 4487$$

$$t_1 = 8019$$

$$t_2 = 1442$$

$$t_3 = 2879$$

$$t_4 = 1657$$

$$p_1 = 1563$$

$$\tilde{a}_4 = 2151$$

$$\tilde{a}_6 = 1624$$

To execute step 310 of Fig. 5B, processing proceeds to step 1310 of Fig. 12. At step 1310 of Fig. 12, the following values are set:

$$p_0 = 1$$

$$p_2 = 1868$$

$$p_3 = 4199$$

$$c_1 = 1571$$

$$c_2 = 2701$$

At step 1320, S is set to S = 3. At step 1330, the following are set:

$$c_3 = 3867$$

$$c_4 = 6078$$

At step 1340, the value $p_4 = 725$ is set. At step 1350, $s_0 = 1$. At step 1360, the following are obtained:

$$s_1 = 1563$$

$$s_2 = 0$$

$$s_3 = 0$$

$$s_4 = 0$$

Processing returns to step 330 of Fig. 5B.

At step 330 of Fig. 5B, since $s_2 = s_3 = s_4 = 0$, processing proceeds to step 360. At step 360, the kernel polynomial is found to be:

$$h(X) = X + 8320$$

To find the eigenvalue e at step 370, processing proceeds to step 905 of Fig. 9.

At step 905, it is determined that the polynomial $h(X)$ is irreducible, that is, it lacks polynomial factors of smaller degree other than constant multiples of itself and 1. After step 910, $h(X) = X + 8320$ is obtained. At step 915, by table look-up, s = 2

is obtained. At step 920, the values for a_2, b_2, c_2 and d_2 from step 1020 are recalled. At step 925, the degree of $h(X)$ is found to be "1", so at step 960, the following values are set:

$$Q_1(X) = 1563$$

$$P_1(X) = 1563$$

$$e = 1$$

At step 965, $P_1(X) = Q_1(X)$ is true, so at step 975, e = 1 is obtained and processing returns to step 380 of Fig. 5B.

At step 380 of Fig. 5B, t is calculated as t = 2. At step 390, F = -1. At step 395, $T_1 = -1$ and $M_1 = 3$.

Continuing to step 400 of Fig. 5C, since $i = i_{max}$ is true, at step 420, g has a value of 2, so the check finds that $2 \neq 36$ and the result is negative. It is noted that, in a practical example, $i_{max} = 70$ is realistic, and so processing would iterate through step 410 $i_{max} - 1 = 69$ times before proceeding to step 420. This is not shown for brevity. Similarly, after the negative result at step 420, processing iterates through step 430 for $\ell = 5, 7, 11, 13, 17, 19$ and 23, in similar manner as described above. Step 380 is executed for $\ell = 13$ and $\ell = 23$. On the next iteration through step 430, processing proceeds to step 170 of Fig. 5A and ℓ is set to $\ell = 29$. To execute step 200, processing proceeds to step 710 of Fig. 7.

At step 710 of Fig. 7, the polynomial $P_{29}(J) = J + 11$ is obtained by table look-up, and the degree ν has a value of 1. To execute step 720, processing proceeds to step 1210 of Fig. 11.

At step 1210 of Fig. 11, the truncated power series X is obtained as:

$$X = q^{-1} + q + q^5 - q^6 - 2q^7 - 2q^{10} + q^{11} - 2q^{15} + q^{19} - 2q^{22} + 2q^{28}$$

$$+ q^{29} + 2\,q^{30} - 2\,q^{31} + 2\,q^{34} + 2\,q^{40} + q^{41} - 2\,q^{42} + 2\,q^{48} - q^{55}$$

At step 1220, k is set to k = -1. At step 1230, a_{-1} is set to the coefficient of q^{-1} in the truncated power series *X*, that is $a_{-1} = 1$. At step 1240, it is checked whether (-1) = (2) (29) (1) -1; since $(-1) \neq 57$, processing proceeds to step 1250 to increment k to k = 0 and return to step 1230. Processing iterates as described above until all the coefficients a_i are determined as follows, all $a_i = 0$ for i = -1 to 57, except:

$$a_{-1} = 1,\ a_1 = 1,\ a_5 = 1,\ a_6 = -1,\ a_7 = -2,\ a_{10} = -2,\ a_{11} = 1,\ a_{15} = -2,$$
$$a_{19} = 1,\ a_{22} = -2,\ a_{28} = 2,\ a_{29} = 1,\ a_{30} = 2,\ a_{31} = -2,\ a_{34} = 2,\ a_{40} = 2,$$
$$a_{41} = 1,\ a_{42} = -2,\ a_{48} = 2,\ a_{55} = -1$$

When k = 57, the test at step 1240 is positive, so processing returns to step 730 of Fig. 7.

At step 730 of Fig. 7, the coefficients b_k are obtained as follows, all $b_k = 0$ for k = -1 to 57 except:

$$b_{-1} = 1,\ b_1 = 1,\ b_5 = 1,\ b_6 = -1,\ b_7 = -2,\ b_{10} = -2,\ b_{11} = 1,\ b_{15} = -2,$$
$$b_{19} = 1,\ b_{22} = -2,\ b_{28} = 2,\ b_{29} = 1,\ b_{30} = 2,\ b_{31} = -2,\ b_{34} = 2,\ b_{40} = 2,$$
$$b_{41} = 1,\ b_{42} = -2,\ b_{48} = 2,\ b_{55} = -1$$

At step 740, the q-series for f is obtained as:

$$f(q) = q^{-1} + 1 + 3q + 4q^2 + 7q^3 + 10q^4 + 17q^5 + 22q^6 + 32q^7 + 44q^8 + 62q^9$$
$$+ 80q^{10} + 112q^{11} + 144q^{12} + 193q^{13} + 248q^{14} + 323q^{15} + 410q^{16}$$
$$+ 530q^{17} + 664q^{18} + 845q^{19} + 1054q^{20} + 1324q^{21} + 1634q^{22}$$
$$+2037q^{23} + 2498q^{24} + 3082q^{25} + 3760q^{26} + 4601q^{27} + 5580q^{28}$$
$$+ 6789q^{29} + 8186q^{30} + 8q^{31} + 1993q^{32} + 4388q^{33} + 7169q^{34} + 627q^{35}$$

$$+ 4494q^{36} + 9110q^{37} + 4575q^{38} + 1025q^{39} + 8356q^{40} + 7125q^{41} + 7218q^{42} + 9059q^{43} + 2813q^{44} + 8730q^{45} + 7152q^{46} + 8581q^{47} + 3277q^{48} + 1895q^{49} + 4675q^{50} + 2655q^{51} + 6093q^{52} + 6263q^{53} + 3636q^{54} + 9551q^{55} + 4936q^{56} + 1411q^{57}$$

At step 750, the power series expansions of $f^2, f^3, \ldots, f^{29}$ are obtained using the q-series expression for f, above. At step 760, the terms $s_k(q)$ are obtained, for example, $s_{16}(q) = 8565 + 457q$. At step 765, the terms $c_k(q)$ are obtained, for example, $c_{11}(q) = 5327 + 89q$. At step 770, the following terms are set:

$$C_1(q) = 9882q^{-1} + 9853 + 776q$$

$$C_{30}(q) = q^{-2} + 8238q^{-1} + 5381$$

At step 775, the terms $C_k(q)$ are obtained, for example, $C_2(q) = 29q^{-1} + 9452$. To execute step 780, processing proceeds to step 810 of Fig. 8.

For brevity, instead of discussing how to obtain all polynomials G_k, only the polynomial G_3 will be discussed. At step 810 of Fig. 8, z is set to $z = C_3(q) = 9564q^{-1} + 8420$. At step 820, t is set to $t = 1$. At step 830, $R = 0$, $b = 1$. At step 840, $R = 9564J$. At step 850, $z = 8564$. At step 860, since $b \neq 0$, processing proceeds to step 870 where b is decremented to $b = 0$, and then returns to step 840. In the second iteration of step 840, $R = 9564J + 8564$. At step 850, $z = 0$. At step 860, $b = 0$, so at step 880, G_3 is set to $G_3 = 9564J + 8564$, and processing returns to step 790 of Fig. 7.

At step 790 of Fig. 7, the modular polynomial Ψ_{29} is computed as:

$$\Psi_{29}(F,J) = F^{30} + (9882J + 714)F^{29} + (29J + 7642)F^{28} + (9564J + 8564)F^{27}$$

$$+ (1421J + 9576)F^{26} + (580J + 2026)F^{25} + (2969J + 729)$$

$$F^{24} + (4264J + 8756)\,F^{23} + (1622J + 6533)\,F^{22} + (231J + 3005)F^{21} + (6003J + 4219)\,F^{20} + (7847J + 4570)\,F^{19} + (4556J + 8942)\,F^{18} + (5613J + 8192)\,F^{17} + (2349J + 1640)\,F^{16} + (4436J + 2545)\,F^{15} + (2625J + 8972)\,F^{14} + (4697J + 861)\,F^{13} + (6155J + 7530)\,F^{12} + (4605J + 2858)\,F^{11} + (2082J + 4883)\,F^{10} + (1815J + 1968)\,F^{9} + (6079J + 2675)\,F^{8} + (118J + 4907)\,F^{7} + (4424J + 9155)\,F^{6} + (1028J + 3410)\,F^{5} + (4890J + 730)\,F^{4} + (3190J + 9362)\,F^{3} + (4727J + 5869)\,F^{2} + (2267J + 1683)\,F + (J^{2} + 6750J + 5409)$$

and processing returns to step 210 of Fig. 5A.

At step 210 of Fig. 5A, i is set to i = 1. The next several iterations are omitted for brevity. For $\ell \in A_1$, processing proceeds through step 380, that is the auxiliary prime ℓ provided information, for ℓ being one of 41, 47, 59, 71, 61, 79, 89, 73, 131, 109, 97, 151, 167, 139 and 137. Discussion of this example resumes with step 440 of Fig. 5C.

At step 440 of Fig. 5C, i is set to i = 1. At step 450, the value M_1

$$M_1 = 150783085059766145035730230806789$$

is compared with $4(9883)^{0.5} = 397.65$. Since M_1 is larger, processing proceeds to step 480, at which g is set to g = 9883 + 1 - 62 = 9822. At step 490, x is found to be x = 6. At step 500, the expression 9822/6 = 1637 is determined to be a prime number. At step 505, it is checked whether 1637 divides $(9883)^k - 1$. Since the result is negative, at step 510, E_1 is determined to be an acceptable elliptic curve.

An example of using an elliptic curve obtained according to the present technique for encryption and decryption will now be discussed.

Let P be a point of prime order q on the curve E{a, b} over the finite field F_p of p elements. Let m be a secret positive integer less than q, m < q, and let G be the point m ⊗ P on E{a, b}, where ⊗ denotes scalar multiplication on the curve. The public key consists of (F_p, E{a, b}, P, q, G) and the private key consists of the integer m.

Encryption and decryption using this public/private key pair may be done as follows. Let M be the message to be encrypted; it is assumed that M is a positive integer smaller than p, the cardinality of F_p, M < p. To encrypt M, choose a random positive integer k less than q and compute the points k ⊗ P and k ⊗ G on the curve E{a, b}. Let k ⊗ G = (x, y). The encryption of M is (k ⊗ P, (x * M) mod p).

To decrypt an encrypted message consisting of the pair (R, S) encrypted according to the encryption method described above where R is a point on the curve and S is a positive integer smaller than p, S < p, the owner of the private key m computes m ⊗ R on the curve E{a, b} using the private key m. Let m ⊗ R = (U, V). The decrypted message is (S/U) mod p.

For the example, with p = 9883, let P = (8508, 3003) be a point of order q =

1637 on the curve E{123, 765}: $Y^2 = X^3 + 123 X + 765$ over $F_{9883} = F_p$. Let m = 1234 be the private key. It follows that m ⊗ P = 1234 ⊗ (8508, 3003) = (4131, 9630) = G, the public point on the curve corresponding to m.

Let M = 1122 be the message to be encrypted. Randomly choose k = 635 and compute

k ⊗ P = 635 ⊗ (8508, 3003) = (4071, 578), and k ⊗ G = 635 ⊗ (4131, 9630) = (5104, 8488). The encryption of M = 1122 is ((4071, 578), (5104 * 1122) mod 9883) = ((4071, 578), 4431).

To decrypt the message (R, S) with R = (4071, 578) and S = 4431, compute m ⊗ R = 1234 ⊗ (4071, 578) = (5104, 8488) = (U, V) with U = 5104. The decrypted message is (S/U) mod p = (4431/5104) mod 9883 = 1122. Note that the resulting decryption is the same as the message M that was encrypted.

An example of using an elliptic curve obtained according to the present technique for generation and verification of digital signatures will now be discussed.

Let P be a point of prime order q on the curve E{a, b} over the finite field F_p of p elements. Let m be a secret positive integer less than q, m < q, and let G be the point m ⊗ P on E{a, b}, where ⊗ denotes scalar multiplication on the curve. The public key consists of

(F_p, E{a, b}, P, q, G) and the private key consists of the integer m.

Generation of a digital signature may be done as follows. Let d be the value of a cryptographically secure hash function applied to the message to be signed. Choose the hash function to assure $0 < d < q$. Pick a random positive integer k, $k < q$. Calculate $k \otimes P = (x, y)$. Calculate $r = (x + d) \bmod q$ and $s = (k - m\, r) \bmod q$. The digital signature for the message of hash value d is the pair (r, s).

Verification of a digital signature (r, s) for a message of hash value d is as follows.

Calculate $s \otimes P + r \otimes G = (x', y')$. If the integers d and $r - x'$ yield the same residue when divided by q, the signature is deemed valid. Otherwise, the signature is rejected.

For the example, with p = 9883, let P = (8508, 3003) be a point of order q = 1637 on the curve E{123, 765}: $Y^2 = X^3 + 123\,X + 765$ over $F_{9883} = F_p$. Let m = 1234 be the private key. It follows that m ⊗ P = 1234 ⊗ (8508, 3003) = (4131, 9630) = G, the public point on the curve corresponding to m.

Let the hash value to be signed by $d = 876$ and let the randomly chosen integer be $k = 101$. Then $k \otimes P = 101 \otimes (8508, 3003) = (7060, 9514)$, therefore $x = 7060$ and $r = (7060 + 876) \bmod 1637 = 1388$. Furthermore, $s = (101 - 1234 * 1388) \bmod 1637 = 1248$. Therefore, the signature is (1388, 1248).

To verify the signature $(r, s) = (1388, 1248)$ for the message with hash value d, calculate $s \otimes P + r \otimes G = (7060, 9514)$, so that $x' = 7060$. The integers $d = 876$ and $r - x' = -5672$ yield the same residue modulo $q = 1637$, namely, the residue 876. Therefore, the signature is accepted as valid.

Although an illustrative embodiment of the present invention, and various modifications thereof, have been described in detail herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to this precise embodiment and the described modifications, and that various changes and further modifications may be effected therein by one skilled in the art without departing from the scope or spirit of the invention as defined in the appended claims.

What is claimed is:

1. A method of selecting an elliptic curve for a cryptosystem, comprising the steps of:

selecting a prime number p defining a field F_p,

selecting a set of candidate elliptic curves E_i over the field F_p,

finding a set of modular polynomials Ψ_ℓ modulo p for a list of candidate auxiliary primes ℓ by a calculation in characteristic p using a stored polynomial P_ℓ,

finding the roots modulo p of the modular polynomials Ψ_ℓ,

generating kernel polynomials $h(X)$ based on the roots of the modular polynomials Ψ_ℓ,

finding an eigenvalue e for one of the kernel polynomials $h(X)$,

obtaining a value t based on the eigenvalue e and the prime number p,

obtaining the number of points of one of the candidate elliptic curves E_i over F_p using the value t to make a determination whether the candidate elliptic curve is sufficiently secure, and

selecting the candidate elliptic curve for the cryptosystem when the determination is that the candidate elliptic curve is sufficiently secure.

2. The method of claim 1, wherein the step of finding is performed without table look-up of the modular polynomials Ψ_ℓ.

3. The method of claim 1, wherein, when the determination is that the candidate elliptic curve is insufficiently secure, the step of comparing is repeated for another of the candidate elliptic curves E_i.

4. The method of claim 1, wherein the list of auxiliary primes is A = {3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 41, 47, 59, 71, 53, 61, 79, 83, 89, 101, 73, 131, 103, 107, 109, 97, 113, 151, 167, 127, 179, 139, 191, 149, 137, 173}.

5. The method of claim 1, wherein the prime number p has about 200 bits.

6. The method of claim 1, wherein the number of points of the selected elliptic curve is a product of a second prime number and a cofactor, the cofactor having up to 5 bits.

7. A method of encrypting a message M, comprising the steps of:

selecting an elliptic curve E according to the method of claim 1;

selecting a point P of prime order q on the selected elliptic curve E over the field of F_p;

selecting a secret positive integer m and a random positive integer k, $m < q$, $k<q$;

obtain the points $k \otimes P$ and $k \otimes (m \otimes P) = (x, y)$ on the curve E; and

obtaining the point $(k \otimes P, (x * M) \bmod p)$ as the encrypted message.

8. A method of obtaining a digital signature for a message M, comprising the steps of:

selecting an elliptic curve E according to the method of claim 1;

selecting a point P of prime order q on the selected elliptic curve E over the field of F_p;

selecting a secret positive integer m and a random positive integer k, m < q, k < q; obtaining a cryptographically secure hash value d between 1 and q - 1 of the message M;

calculating $k \otimes P = (x, y)$; and

obtaining the pair $((x + d) \bmod q, (k - m (x + d) \bmod q)$ as the digital signature.

9. A portable device for encoding information using an elliptic curve cryptosystem, comprising:

means for selecting an elliptic curve by finding the roots of modular polynomials Ψ_ℓ modulo p for a list of candidate auxiliary primes ℓ and a prime number p by a calculation in characteristic p using a stored polynomial P_ℓ, and

means for encoding the information using the selected elliptic curve.

10. The device of claim 9, further comprising means for decoding received information using the selected elliptic curve.

11. A portable device for digitally signing information using an elliptic curve cryptosystem, comprising:

means for selecting an elliptic curve by finding the roots of modular polynomials Ψ_ℓ modulo p for a list of candidate auxiliary primes ℓ and a prime number p by a calculation in characteristic p using a stored polynomial P_ℓ, and

means for digitally signing the information using the selected elliptic curve.

12. The device of claim 11, further comprising means for verifying a received digital signature using the selected elliptic curve.

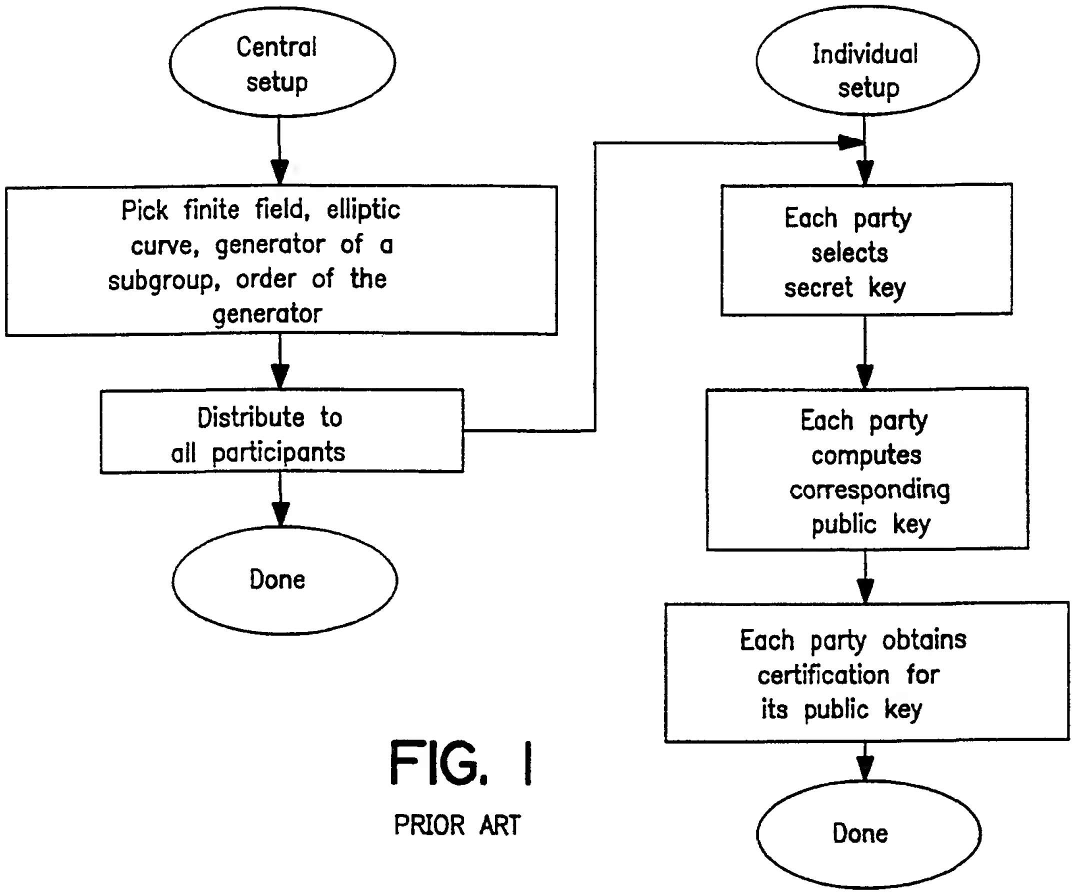


FIG. 1

PRIOR ART

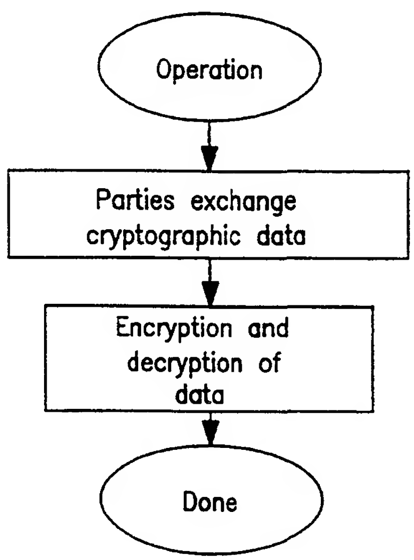


FIG. 2

PRIOR ART

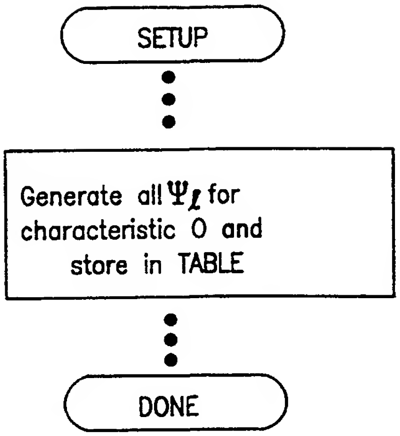


FIG. 3A

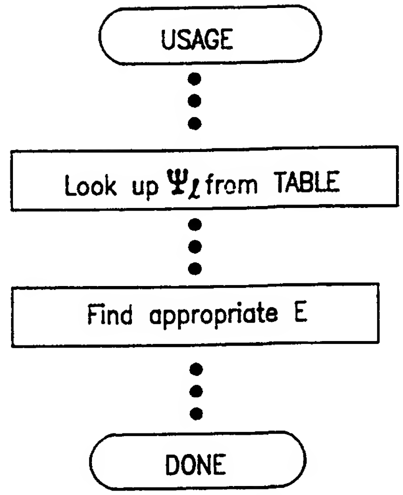


FIG. 3B

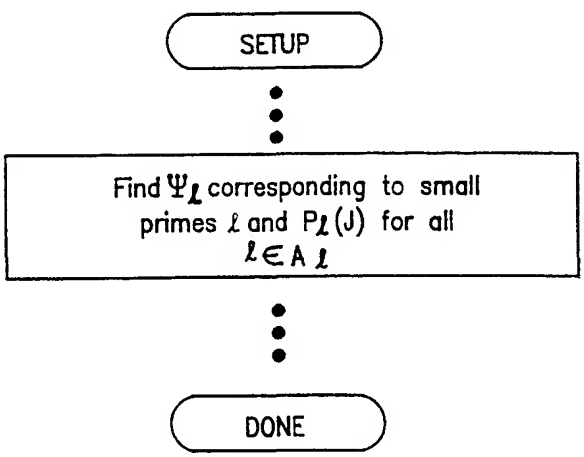


FIG. 4A

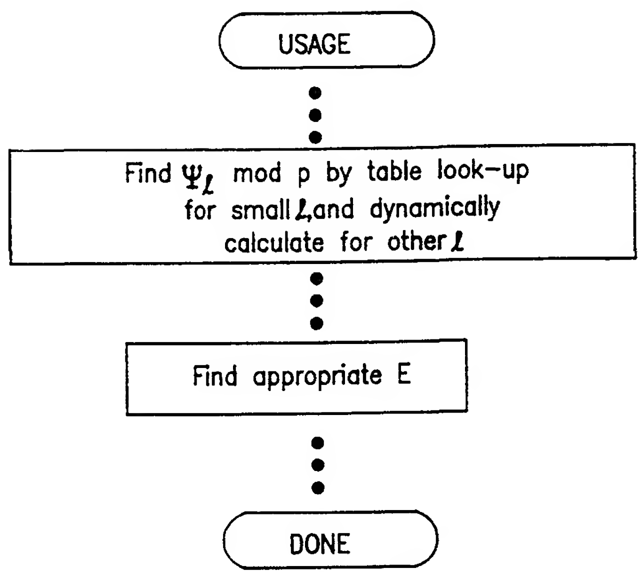


FIG. 4B

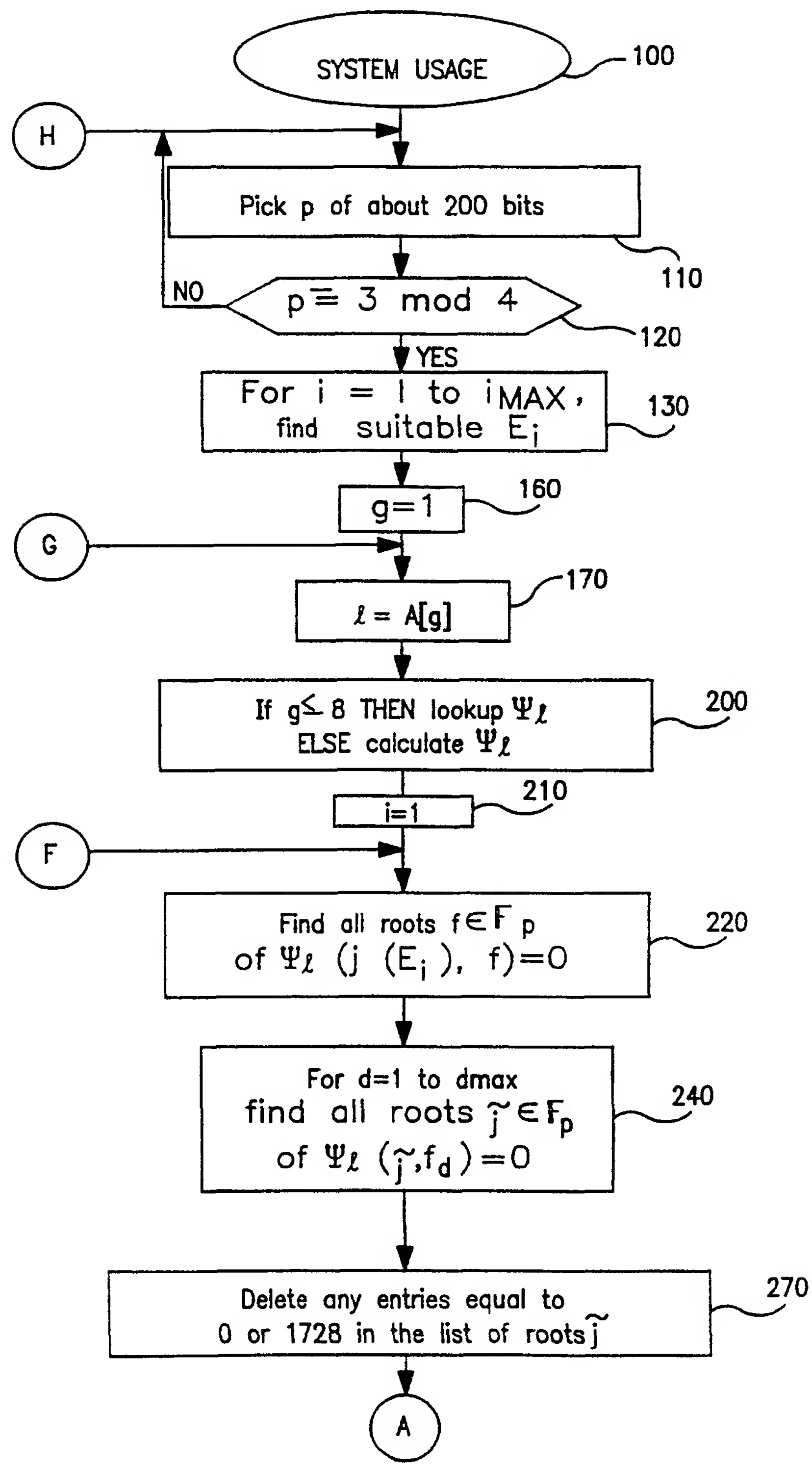

SYSTEM USAGE
100
H
Pick p of about 200 bits
110
NO
p ≡ 3 mod 4
120
YES
For i = 1 to iMAX, find suitable Ei
130
g=1
160
G
ℓ = A[g]
170
If g ≤ 8 THEN lookup Ψℓ ELSE calculate Ψℓ
200
i=1
210
F
Find all roots f ∈ F p of Ψℓ (j (Ei), f)=0
220
For d=1 to dmax find all roots j̃ ∈ Fp of Ψℓ (j̃, fd)=0
240
Delete any entries equal to 0 or 1728 in the list of roots j̃
270
A

FIG. 5A

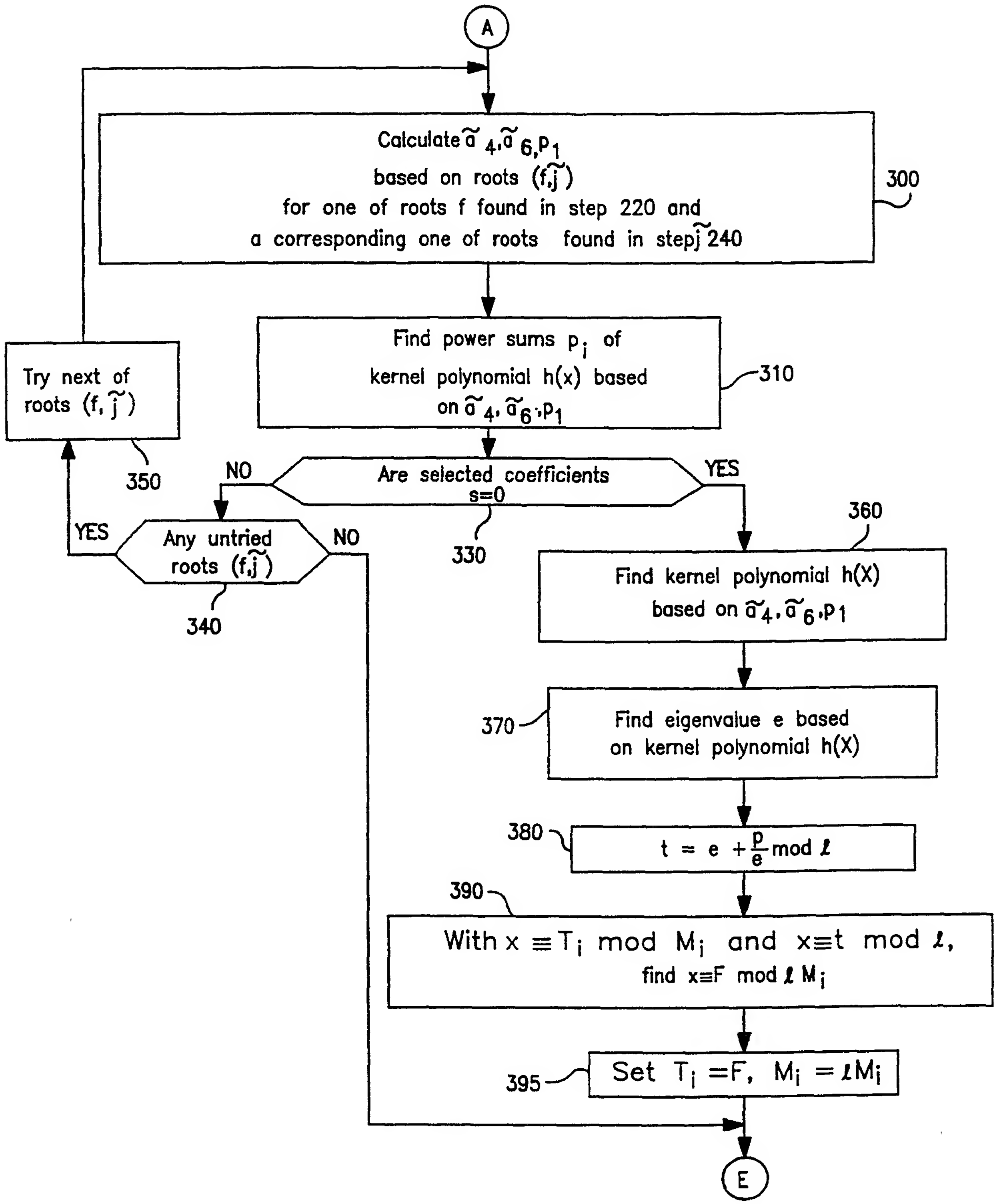

A
Calculate $\tilde{a}_4, \tilde{a}_6, P_1$ based on roots $(f, \tilde{j})$ for one of roots f found in step 220 and a corresponding one of roots $\tilde{j}$ found in step 240
300
Find power sums p_i of kernel polynomial h(x) based on $\tilde{a}_4, \tilde{a}_6, P_1$
310
Are selected coefficients s=0
330
NO
YES
Any untried roots $(f, \tilde{j})$
340
YES
NO
Try next of roots $(f, \tilde{j})$
350
Find kernel polynomial h(X) based on $\tilde{a}_4, \tilde{a}_6, P_1$
360
Find eigenvalue e based on kernel polynomial h(X)
370
$t = e + \frac{p}{e} \bmod \ell$
380
With $x \equiv T_i \bmod M_i$ and $x \equiv t \bmod \ell$, find $x \equiv F \bmod \ell M_i$
390
Set $T_i = F$, $M_i = \ell M_i$
395
E

FIG. 5B

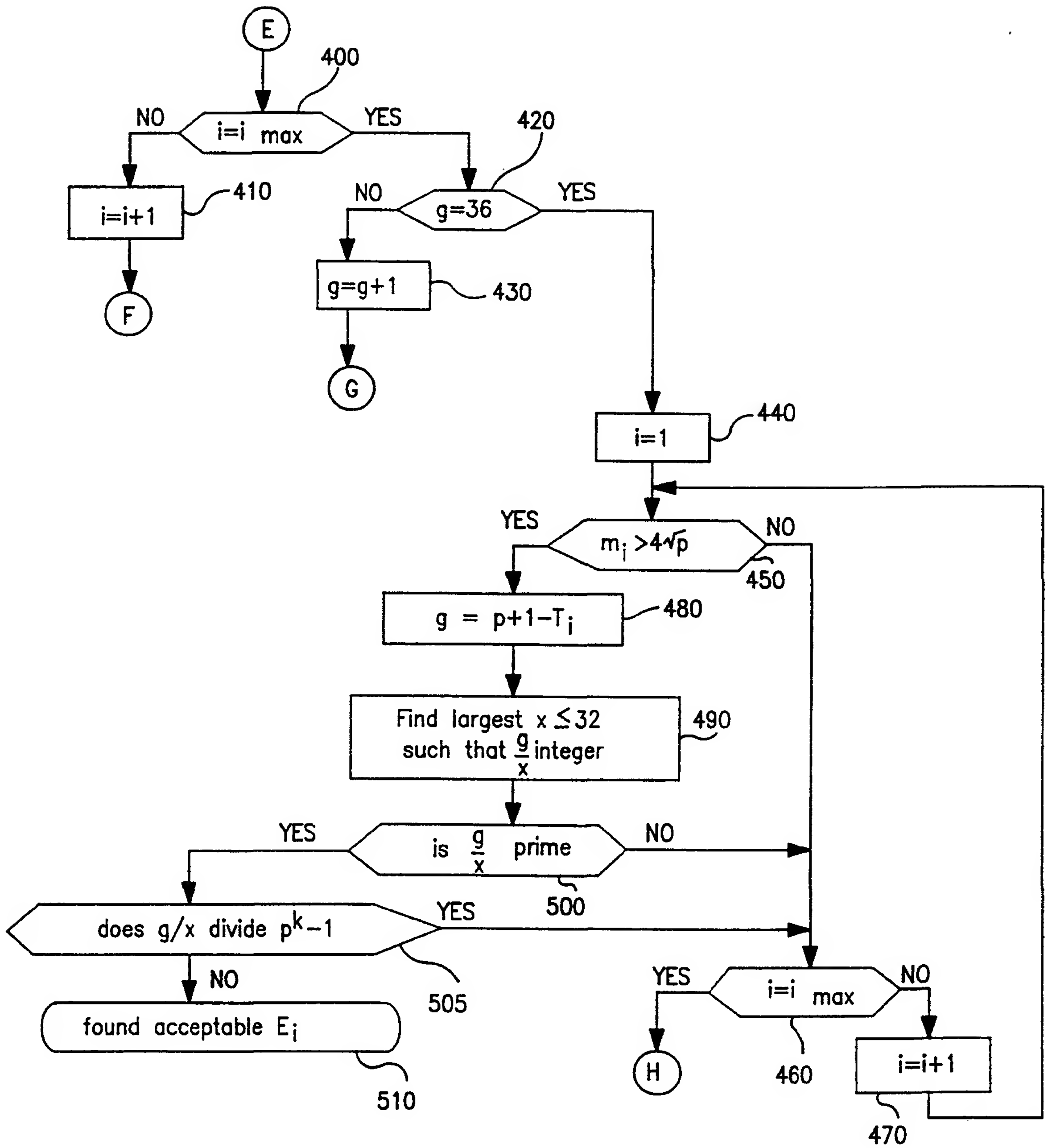

E
400
NO
i=i max
YES
420
i=i+1
410
NO
g=36
YES
F
g=g+1
430
G
i=1
440
YES
$m_i > 4\sqrt{p}$
NO
450
$g = p+1-T_i$
480
Find largest $x \leq 32$ such that $\frac{g}{x}$ integer
490
YES
is $\frac{g}{x}$ prime
NO
500
does g/x divide p^k-1
YES
NO
505
found acceptable E_i
510
YES
i=i max
NO
H
460
i=i+1
470

FIG. 5C

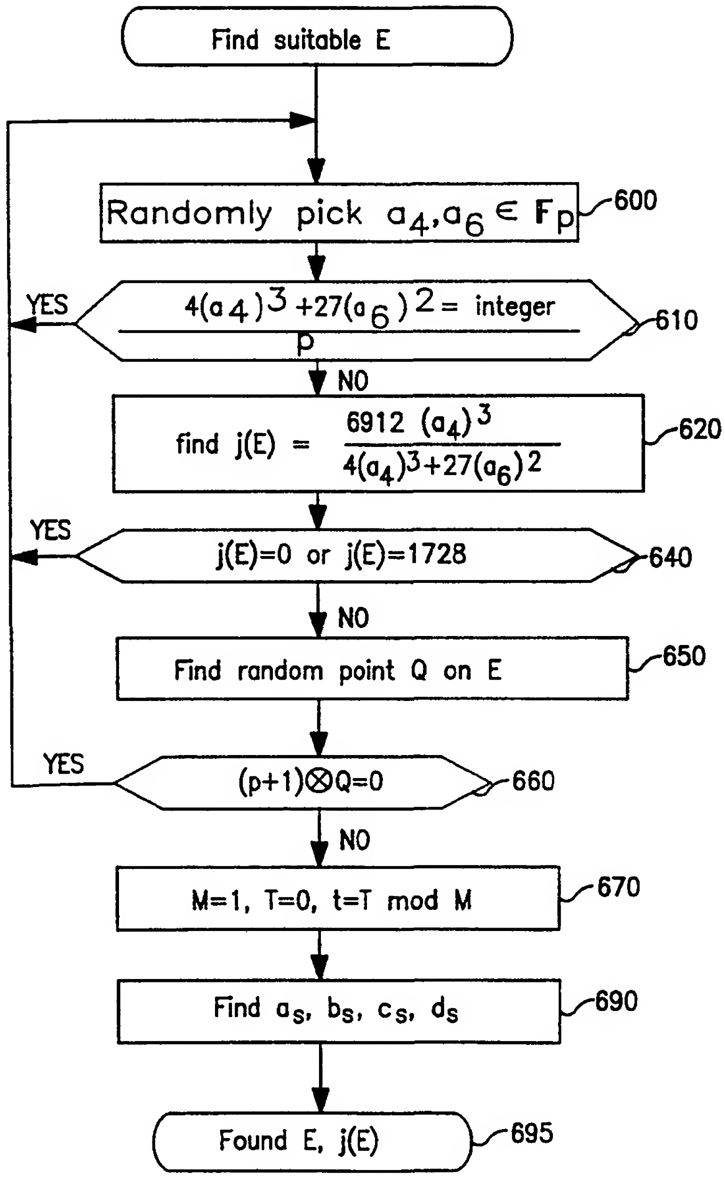

Find suitable E
Randomly pick $a_4, a_6 \in \mathbf{F}_p$
600
$\frac{4(a_4)^3+27(a_6)^2}{p}$ = integer
610
YES
NO
find $j(E) = \frac{6912\,(a_4)^3}{4(a_4)^3+27(a_6)^2}$
620
j(E)=0 or j(E)=1728
640
YES
NO
Find random point Q on E
650
$(p+1)\otimes Q=0$
660
YES
NO
M=1, T=0, t=T mod M
670
Find a_s, b_s, c_s, d_s
690
Found E, j(E)
695

FIG. 6

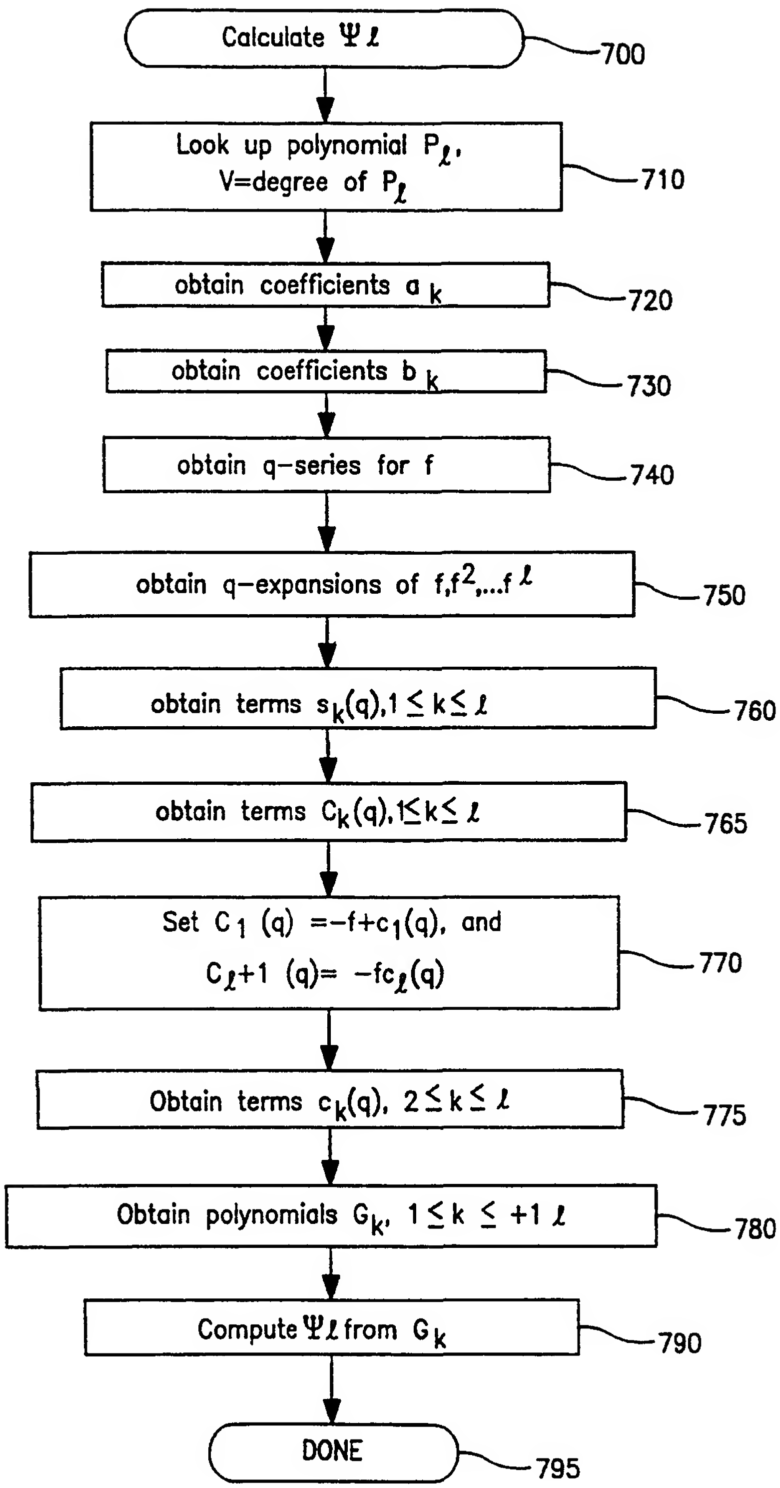
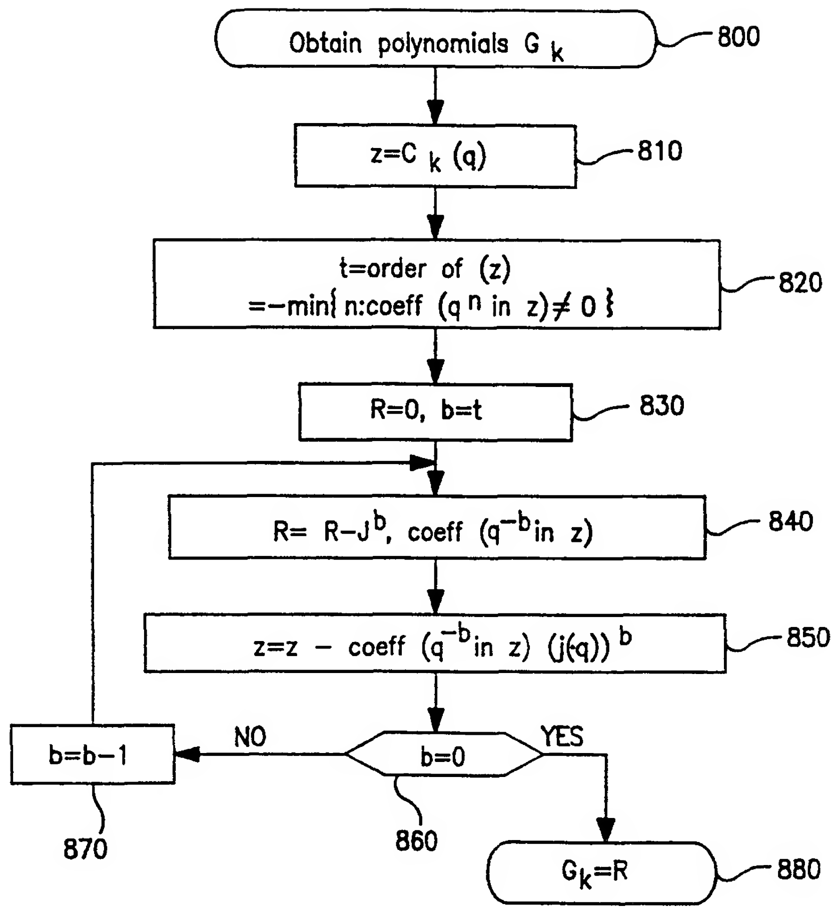

Calculate Ψ_ℓ
700
Look up polynomial P_ℓ, V=degree of P_ℓ
710
obtain coefficients a_k
720
obtain coefficients b_k
730
obtain q-series for f
740
obtain q-expansions of $f, f^2, \ldots f^\ell$
750
obtain terms $s_k(q), 1 \leq k \leq \ell$
760
obtain terms $C_k(q), 1 \leq k \leq \ell$
765
Set $C_1(q) = -f + c_1(q)$, and $C_\ell + 1\ (q) = -fc_\ell(q)$
770
Obtain terms $c_k(q),\ 2 \leq k \leq \ell$
775
Obtain polynomials G_k, $1 \leq k \leq +1\ \ell$
780
Compute Ψ_ℓ from G_k
790
DONE
795

FIG. 7



Obtain polynomials G k
800
z=C k (q)
810
t=order of (z)
=−min{ n:coeff (q n in z)≠ 0 }
820
R=0, b=t
830
R= R−J b, coeff (q−b in z)
840
z=z − coeff (q−b in z) (j(q)) b
850
b=b−1
NO
b=0
YES
870
860
Gk=R
880

FIG. 8

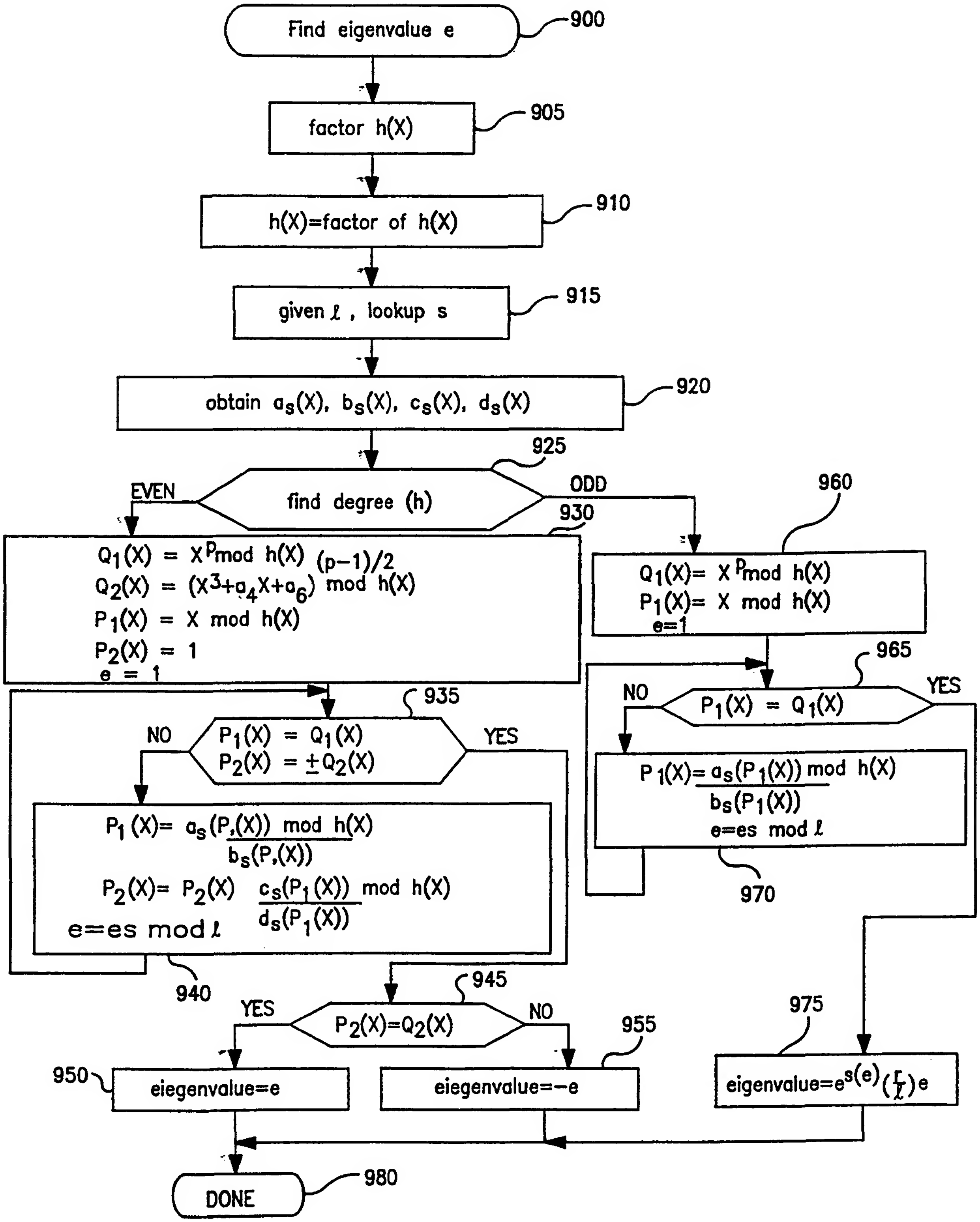

Find eigenvalue e
900
factor h(X)
905
h(X)=factor of h(X)
910
given ℓ, lookup s
915
obtain $a_s(X)$, $b_s(X)$, $c_s(X)$, $d_s(X)$
920
find degree (h)
925
EVEN
ODD
930
960
$Q_1(X) = X^p \bmod h(X)$
$Q_2(X) = (X^3+a_4X+a_6)^{(p-1)/2} \bmod h(X)$
$P_1(X) = X \bmod h(X)$
$P_2(X) = 1$
e = 1
$Q_1(X)= X^p \bmod h(X)$
$P_1(X)= X \bmod h(X)$
e=1
935
NO
$P_1(X) = Q_1(X)$
$P_2(X) = \pm Q_2(X)$
YES
$P_1(X)= \frac{a_s(P_1(X))}{b_s(P_1(X))} \bmod h(X)$
$P_2(X)= P_2(X) \frac{c_s(P_1(X))}{d_s(P_1(X))} \bmod h(X)$
e=es mod ℓ
940
965
NO
$P_1(X) = Q_1(X)$
YES
$P_1(X)= \frac{a_s(P_1(X))}{b_s(P_1(X))} \bmod h(X)$
e=es mod ℓ
970
945
YES
$P_2(X)=Q_2(X)$
NO
955
975
950
eiegenvalue=e
eiegenvalue=−e
eigenvalue=$e^{s(e)}\left(\frac{r}{\ell}\right)e$
DONE
980

FIG. 9

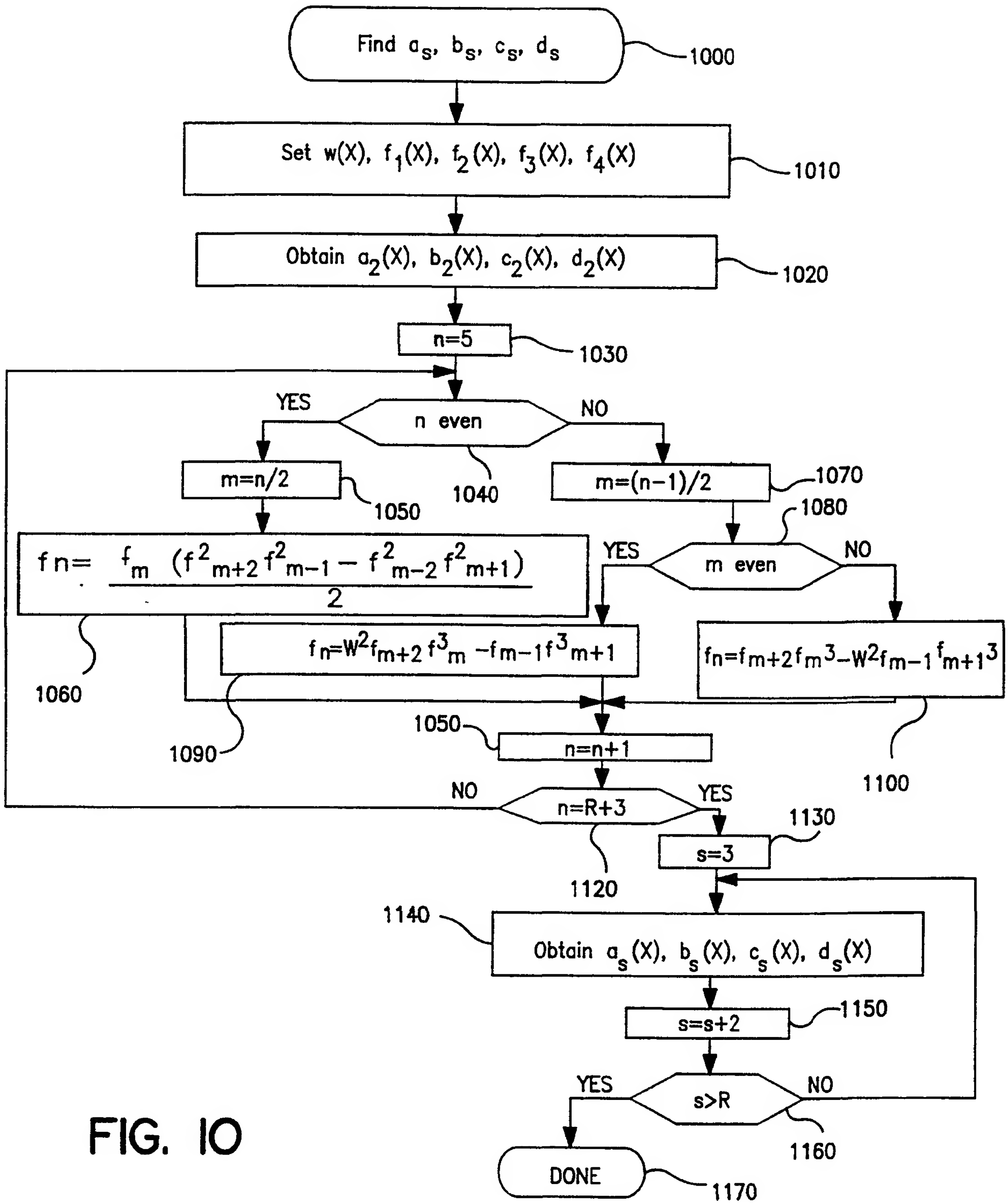

Find a_s, b_s, c_s, d_s
1000
Set $w(X)$, $f_1(X)$, $f_2(X)$, $f_3(X)$, $f_4(X)$
1010
Obtain $a_2(X)$, $b_2(X)$, $c_2(X)$, $d_2(X)$
1020
n=5
1030
YES
n even
NO
1040
m=n/2
1050
m=(n-1)/2
1070
1080
YES
m even
NO
$f_n = \frac{f_m (f^2_{m+2} f^2_{m-1} - f^2_{m-2} f^2_{m+1})}{2}$
1060
$f_n = W^2 f_{m+2} f^3_m - f_{m-1} f^3_{m+1}$
1090
$f_n = f_{m+2} f_m^3 - W^2 f_{m-1} f_{m+1}^3$
1100
1050
n=n+1
NO
n=R+3
YES
1120
s=3
1130
1140
Obtain $a_s(X)$, $b_s(X)$, $c_s(X)$, $d_s(X)$
s=s+2
1150
YES
s>R
NO
1160
DONE
1170

FIG. 10

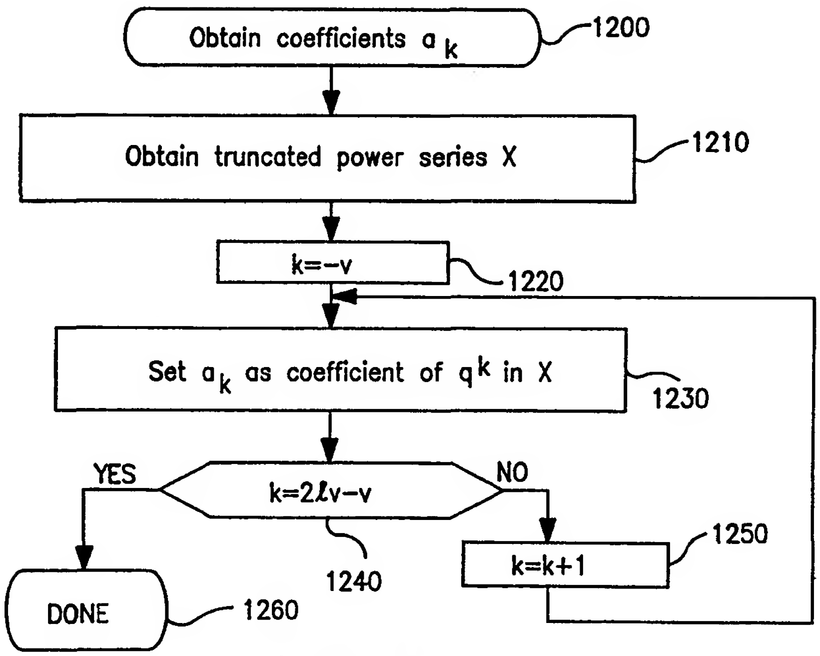

Obtain coefficients a k
1200
Obtain truncated power series X
1210
k=−v
1220
Set a k as coefficient of q k in X
1230
YES
k=2ℓv−v
NO
1240
k=k+1
1250
DONE
1260

FIG. 11

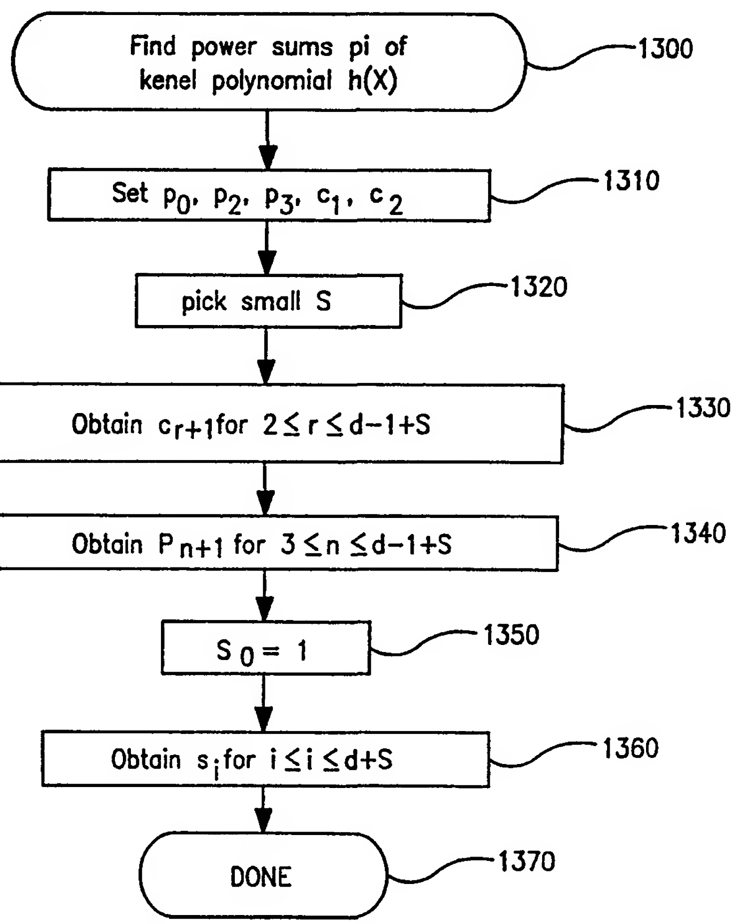

Find power sums pi of kenel polynomial h(X)
1300
Set p0, p2, p3, c1, c2
1310
pick small S
1320
Obtain cr+1 for 2≤r≤d−1+S
1330
Obtain Pn+1 for 3≤n≤d−1+S
1340
S0 = 1
1350
Obtain si for i≤i≤d+S
1360
DONE
1370

FIG. 12

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/20411

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L9/30 G06F7/72

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication where appropriate, of the relevant passages	Relevant to claim No
A	US 5 442 707 A (MIYAJI ATSUKO ET AL) 15 August 1995 (1995-08-15) column 10, line 5 - line 46 column 11, line 8 - line 48 column 13, line 31 - line 38 ---	1,8,9,11

-/--

[X] Further documents are listed in the continuation of box C

[X] Patent family members are listed in annex

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance, the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 January 2000	04/02/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P B 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel (+31–70) 340–2040, Tx 31 651 epo nl, Fax: (+31–70) 340–3016	Holper, G

1

INTERNATIONAL SEARCH REPORT

In tional Application No
PCT/US 99/20411

C (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication,where appropriate, of the relevant passages	Relevant to claim No
A	IZU T ET AL: "Parameters for secure elliptic curve cryptosystem-improvements on Schoof's algorithm" PUBLIC KEY CRYPTOGRAPHY. FIRST INTERNATIONAL WORKSHOP ON PRACTICE AND THEORY IN PUBLIC KEY CRYPTOGRAPHY, PKC'98. PROCEEDINGS, PUBLIC KEY CRYPTOGRAPHY FIRST INTERNATIONAL WORKSHOP ON PRACTICE AND THEORY IN PUBLIC KEY CRYPTOGRAPHY, PKC'98 PROCEEDINGS, , 5 February 1998 (1998-02-05), pages 253-257, XP000870397 1998, Berlin, Germany, Springer-Verlag, Germany ISBN: 3-540-64693-0 abstract page 254, paragraph 2 -----	1

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

In :tional Application No
PCT/US 99/20411

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5442707 A	15-08-1995	JP 6110386 A	22-04-1994
		JP 6295154 A	21-10-1994